Improving Information Extraction and Translation Using Component Interactions

by

Heng Ji

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

January, 2008

Approved: __________________________

Prof. Ralph Grishman

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **JAN 2008**	2. REPORT TYPE	3. DATES COVERED **00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE **Improving Information Extraction and Translation Using Component Interactions**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **New York University,Department of Computer Science,251 Mercer Street ,New York,NY,10012**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The traditional natural language processing pipeline incorporates multiple stages of linguistic analysis. Although errors are typically compounded through the pipeline, it is possible to reduce the errors in one stage by harnessing the results of the other stages. This thesis presents a new framework based on component interactions to approach this goal. The new framework applies all stages in a suitable order, with each stage generating multiple hypotheses and propagating them through the whole pipeline. Then the feedback from subsequent stages is used to enhance the target stage by re-ranking these hypotheses, and then produce the best analysis. The effectiveness of this framework has been demonstrated by substantially improving the performance of Chinese and English entity extraction and Chinese-to-English entity translation. The inference knowledge includes mono-lingual interactions among information extraction stages such as name tagging, coreference resolution, relation extraction and event extraction, as well as cross-lingual interaction between information extraction and machine translation. Such symbiosis of analysis components allows us to incorporate information from a much wider context, spanning the entire document and even going across documents, and utilize deeper semantic analysis; it will therefore be essential for the creation of a highperformance NLP pipeline.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**163**	

Dedicated to my grandmother Lv Aixiang.

谨献给我的外婆吕爱香, 祝福她健康快乐。

ACKNOWLEDGEMENTS

Firstly and most importantly, I am eternally grateful to my advisor, Prof. Ralph Grishman, for his tremendous guidance, great advice, engaging teaching and lots of ideas throughout my PhD study. I could not have imagined having a more excellent advisor and mentor. With his enthusiasm, his inspiration, his perceptiveness, his preciseness, and his great efforts to explain things clearly and simply, he helped to make computational linguistics from 'interesting' to 'fantastic' for me. In addition he has provided me the greatest opportunities to participate in large-scale project evaluations, through which I learned how to develop and manage research projects effectively. ‘Grishman’s student’ is the best ‘nominal mention’ I have been referring to at academic conferences and I feel extremely proud of. Furthermore he provided the most powerful support and encouragements when I faced an algorithm exam disaster or a paper rejection notice. I am really glad that I have come to get know Ralph in my life and will always regard him as my academic idol.

I would like to say a big 'thank-you' to Prof. Satoshi Sekine who kept an eye on the progress of my research work. His critical comments and encouragements were always stimulating and recalling my interests in NLP research. I especially wish to thank Prof. I. Dan Melamed for his nice advice and help since the first time I arrived at this country, and for providing me many presentation opportunities to build my self-confidence.

I would like to thank the other two members of my PhD committee, Dr. Radu Florian and Dr. Kishore Papineni, who took most valuable time in reading my thesis and providing me with extensive comments on my research and career planning.

I would like to say many thanks to Dr. Adam Meyers for providing me many linguistic guidance and resources. He has taught me to conduct linguistics-oriented NLP research. I am indebted to many other present and past Proteus members for providing a fun environment in which to learn and grow. I especially cherish the fun days I study together with Yusuke Shinyama and Cristina Mota, and thank Kiyoshi Sudo and Roman Yangarber for introducing me to industrial and academic communities.

I would like to thank Prof. Mehryar Mohri and my friend Dr. Cynthia Rudin for teaching me the machine learning models used in this thesis.

I also would like to thank other researchers at IBM T.J. Watson Research Center, especially Dr. Salim Roukos, Dr. Nanda Kambhatla, Dr. Imed Zitouni and Dr. Xiaoqiang Luo for their wonderful mentoring and help through my internship.

I thank the colleagues in the NIGHTINGALE team, especially Prof. Mari Ostendorf, Prof. Mary Harper, Prof. Hermann Ney, Dr. Dilek Hakkani-Tur and Dr. Jordan Cohen for their wise advice and help. I particularly enjoyed working with entity translation researchers at Fair Isaac Corp. and RWTH Aachen: Matthias Blume, John Wang, Shahram Khadivi, Richard Zens and Yuqi Zhang. I cherish my friendship with Dr. Wen Wang at SRI International and Dr. Yang Liu at UT Dallas.

It was my former advisor Prof. Zhensheng Luo who led me into the NLP realm when I was a first-year undergraduate student many years ago. I deeply appreciate his paternal words, emails and oversea phone calls whenever I wanted to give up. I thank him for telling me NLP is a more than wonderful field which is worth spending my whole life to explore.

I am most grateful for my husband and best friend Xing, for his caring, patience, support and love. I also wish to thank my grandparents, my parents, my parents-in-law, my uncles and my younger sister for always making me feel like I could do everything best. I thank all my friends in New York and Beijing for their support and care through my PhD life. Finally, my most special thanks go to my grandmother for bringing me up and teaching me to bravely face difficulties and sadness.

This thesis is supported by the Defense Advanced Research Projects Agency under Contract No. HR0011-06-C-0023, and the National Science Foundation under Grant IIS-00325657.

ABSTRACT

The traditional natural language processing pipeline incorporates multiple stages of linguistic analysis. Although errors are typically compounded through the pipeline, it is possible to reduce the errors in one stage by harnessing the results of the other stages.

This thesis presents a new framework based on component interactions to approach this goal. The new framework applies all stages in a suitable order, with each stage generating multiple hypotheses and propagating them through the whole pipeline. Then the feedback from subsequent stages is used to enhance the target stage by re-ranking these hypotheses, and then produce the best analysis.

The effectiveness of this framework has been demonstrated by substantially improving the performance of Chinese and English entity extraction and Chinese-to-English entity translation. The inference knowledge includes mono-lingual interactions among information extraction stages such as name tagging, coreference resolution, relation extraction and event extraction, as well as cross-lingual interaction between information extraction and machine translation.

Such symbiosis of analysis components allows us to incorporate information from a much wider context, spanning the entire document and even going across documents, and utilize deeper semantic analysis; it will therefore be essential for the creation of a high-performance NLP pipeline.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

NLP systems are typically organized as a pipeline architecture of processing stages (e.g. from speech recognition, to source language information extraction, to machine translation, to target language information extraction and summarization). Each of these stages has been studied separately and quite intensively over the past decade. Annotated corpora have been prepared for each stage, a wide range of models and machine learning methods have been applied, and separate official evaluations have been organized. There has clearly been a great deal of progress on some of these components.

However, the output of each stage is chosen locally and passed to the next step, and there is no feedback from later stages to earlier ones. Although this makes the systems comparatively easy to assemble, it comes at a high price: errors accumulate as information progresses through the pipeline, and an error once made cannot be corrected.

Therefore in the NLP community there has been increasing interest in moving away from systems that make chains of independent local decisions, and instead toward systems that make multiple decisions jointly using global information. This thesis will focus on using stage interactions to improve the task of Information Extraction (IE). Section 1.1 will introduce the IE task, and section 1.2 presents the limitations of two traditional IE frameworks, then in section 1.3 a new IE framework is presented.

1.1 Information Extraction

Information Extraction (IE), at the heart of many natural language processing (NLP) applications, is a task of identifying important types of facts (entities, relations and events) in unstructured text. IE systems typically include name identification and classification, parsing (or partial parsing), semantic classification of noun phrases, coreference resolution, relation extraction and event extraction. Named entity, coreference and template element are reflected in the evaluation tasks introduced for MUC-6 (Grishman and Sundheim, 1996), and template relation was introduced in MUC-7. These tasks have been introduced again in NIST ACE (Automatic Content Extraction)[1] evaluations, with more specific types of entities/relations/events defined.

Placing IE in the multi-lingual NLP environment, source language IE may help a machine translation (MT) system to translate important facts more accurately; while target language IE can provide the knowledge base for information retrieval, question answering and text summarization.

1.2 Traditional IE Frameworks and Their Limitations

In a *sequential* IE framework, various analysis components are arranged sequentially to preprocess text for extraction, with each stage depending on the results of several prior stages and generating a single hypothesis (Figure 1-1); for example, coreference depends on name identification, and event detection depends on parsing. This provides a simple modular organization for the extraction components. For instance, the top ACE systems such as BBN system (Boschee et al., 2005), IBM Cascade Model (Florian et al., 2006) and NYU system (Grishman, 2004; Grishman et al., 2005) were developed in this sequential style.

[1] The ACE task description can be found at http://www.itl.nist.gov/speech/tests/ace/, and the ACE guidelines at http://www.ldc.upenn.edu/Projects/ACE/

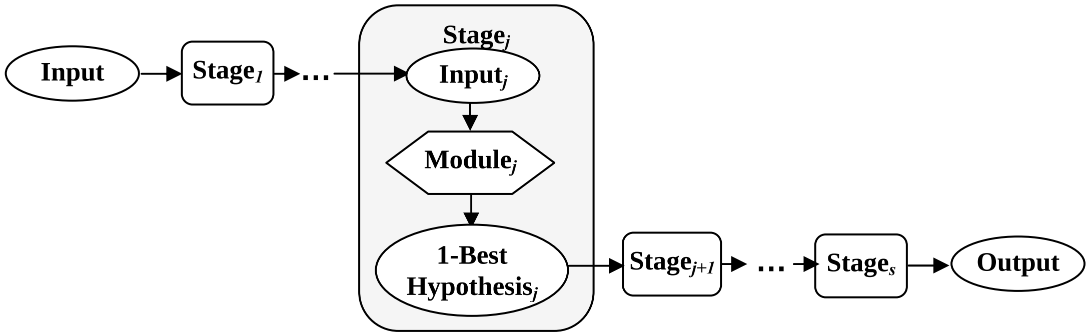


Figure 1-1. Sequential IE Framework

Unfortunately, this sequential organization means that the error rate of the final (combined) analysis grows as each stage also introduces a certain level of its own errors into the analysis. For example, errors in name recognition may lead to errors in reference resolution. Indeed, because of the interdependence of the stages, the errors are often compounded from stage to stage. As a net result, the overall system performance can be quite poor even if the individual stages seem satisfactory. The 60% - 70% IE performance barrier has been notoriously difficult to break through.

One limitation is the relatively local features employed by the early stages. In a global view, subsequent stages can often aid in resolving decisions which were difficult for prior stages. Most name taggers, for example, are based on simple models that look only one or two tokens ahead and behind. This fails to capture such basic tendencies as the increased likelihood of a name mentioned once in a document to be mentioned again (knowledge from reference resolution); an employee of some organization is likely to be a person name (knowledge from relation detection).

But such interactions are not easily exploited in a simple sequential model. To account for this, some systems employ a name cache or, more elaborately, features based on the context of other instances of the same string (Chieu and Ng 2002) – in effect, trying to do simple coreference within the name tagger. However, preferences which depend on more complex syntactic structures – for instances, that names involving some particular events are likely to be person names – remain difficult to capture because the event structures are simply not available at this stage of analysis. It may even be difficult to use this information implicitly, by using features which are also used in later stages, because the representation used in the initial stages is too limited.

To address these limitations, some recent systems have used more parallel designs (Figure 1-2), in which a single master classifier incorporates a wide range of features representing the information to address several separate stages (Florian et al., 2004, Zelenko et al., 2004; Daume III and Marcu, 2005). This thesis will refer to such designs as *monolithic* IE frameworks.

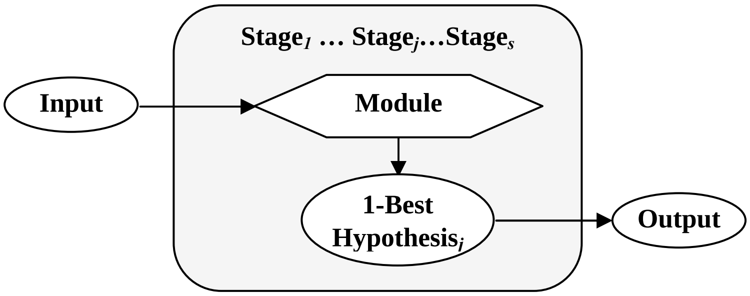


Figure 1-2. Monolithic IE Framework

This can reduce the compounding of errors of the sequential design. However, it leads to a very large feature space and makes it difficult to select linguistically appropriate features for particular analysis tasks. And the constructs created in earlier stages cannot be used in later stages. Furthermore, because these decisions are made at the same time, it becomes much harder to express the stage interactions.

1.3 Stage Interaction based IE Framework

The goal of the thesis is to combine the advantages of sequential and monolithic IE frameworks while overcoming their weaknesses. As already discussed, extraction will never be perfectly accurate, and some of the most problematic consequences of this occur when the final answer is the result of a cascade of processing steps, through which errors accumulate. Information extraction is a potentially symbiotic pipeline with strong dependencies between stages. This thesis transforms this problem – the varied interaction between stages of analysis – into a benefit by exploiting the interactions to reduce the errors in individual stages.

In order to capture these interactions more explicitly, we employ a more general framework which harnesses the richer representations of the later stages to aid the performance of earlier ones. This new framework keeps the sequential design, generates multiple hypotheses and forwards them from each stage to the next, and then uses information from subsequent subtasks to re-rank these hypotheses. Then the top hypothesis after re-ranking is generated as the system output. In doing so, the new framework holds the following advantages:

- ❑ The performance of components which come early in the pipeline and use primarily local knowledge, such as name tagging, can be significantly improved by the much broader context.

- Rather than having errors accumulate, this approach can actually use the feedback from later processing steps to correct the errors from earlier steps.
- The feedback from stages of deeper semantic analysis such as relation detection can help coreference resolution.
- The interaction analysis is extended to cross-lingual level, so that IE and MT can indirectly share the valuable training resources.

There are two areas of focus in this thesis:

- Exploring specific interactions between different analysis levels: the mono-lingual interaction between stages within IE; the cross-lingual interaction between IE and machine translation.
- Effectively organizing these interactions using heuristic rules or supervised re-ranking models.

1.4 Conclusion

The decomposition of NLP systems into components and the intensive study of individual components have been crucial to the advances in NLP. But it is important not to lose sight of the fact that the analysis of a discourse is ultimately a unified process, with a goal of obtaining the most coherent interpretation consistent with the information explicitly expressed, and that this overall goal is reflected in the interactions of the individual components. Understanding these individual interactions can lay the groundwork for an improved NLP pipeline. The rest of this thesis is structured as follows.

Chapter 2: describes our main research task and general setting.

Chapter 3: describes some previous work using global interaction knowledge and an N-Best hypothesis framework.

Chapter 4: presents the baseline information extraction and translation pipeline we developed for the thesis.

Chapter 5: presents diverse types of linguistic interactions for IE.

Chapter 6: describes the general stage-interaction framework for IE, and explains in detail the various algorithms for using interactions.

Chapter 7 and 8: present three case studies of mono-lingual and cross-lingual stage interactions.

Chapter 9: briefly presents some recent relevant publications which cited the work in this thesis.

Chapter 10: concludes the thesis and sketches the future work.

2. TERMINOLOGY

In order to study the stage interactions, this thesis chose to organize and conduct experiments within the context of the ACE evaluations, a specific information extraction task. The remaining chapters will use the following ontology created by ACE to explain the central ideas.

2.1 Entity Detection and Recognition

ACE defines the following terminologies for the entity detection and recognition task:

entity: an object or a set of objects in one of the semantic categories of interest

mention: a reference to an entity (typically, a noun phrase)

name mention: a reference by name to an entity

nominal mention: a reference by a common noun or noun phrase to an entity

The 2005 ACE evaluation had 7 types of entities: PER (persons), ORG (organizations), GPE ('geo-political entities' – locations which are also political units, such as countries, counties, and cities), LOC (other locations without governments, such as bodies of water and mountains), FAC (facility), WEA (Weapon) and VEH (Vehicle). For example, in the following sentence:

Fred Smith became the new prime minister.

There are two mentions:

Mention$_1$_extent = "Fred Smith", Mention$_1$_head = "Fred Smith",

Mention$_1$_entity type = "PER", Mention$_1$_mention type = "NAME";

Mention$_2$_extent = "the new prime minister", Mention$_2$_head = "minister", Mention$_2$_entity type = “PER”, Mention$_2$_mention type = “NOMINAL”.

2.2 Relation Detection

Relation detection involves finding a specified set of relationships between a pair of entities. ACE 2005 had 6 types of semantic relations, with 18 subtypes. The following table lists an example for each relation type.

Relation Type	Example
Agent-Artifact (User-Owner-Inventor-Manufacturer)	Rubin Military Design, the **makers** of the **Kursk**
ORG-Affliation (Employment)	Mr. Smith, the **CEO** of **Microsoft**
Gen-Affiliation (Citizen-Resident-Religion-Ethnicity)	**Salzburg** Red Cross **officials**
Personal-Social (Family)	**Fred**’s **wife**
Part-Whole (Subsidiary)	The **U.S. Congress**
Physical (Near)	a **town** some 50 miles south of **Salzburg**

Table 2-1. Examples of the ACE Relation Types

2.3 Event Recognition

ACE 2005 defined 8 types of events, with 33 subtypes. Table 2-2 provides some ACE event examples.

Event Type	Example
Life (Die)	**Kurt Schork** died in **Sierra Leone yesterday**
Transaction (Transfer)	**GM sold the company** in **Nov 1998** to **LLC**
Movement (Transport)	**Homeless people** have been **moved** to **schools**
Business (Start-ORG)	**Schweitzer founded a hospital** in **1913**
Conflict (Attack)	The **attack** on **Gaza** killed **13 people**
Contact (Meet)	**Arafat’s cabinet met** for **4 hours**
Personnel (Start-Position)	**Cornell Medical Center recruited 12 nursing students**
Justice (Arrest)	**Zawahiri was arrested** in **Iran**

Table 2-2. Examples of the ACE Event Types

2.4 Entity Translation

There was a cross-lingual IE track at ACE 2007 – entity translation (ET)[2] which required systems to take in a text document in a foreign language (e.g. Chinese or Arabic) and extract English language catalog of the entities mentioned in the document.

2.5 Conclusion

Over the year (through 2007), ACE has included evaluations on four languages: English, Chinese, Arabic and most recently Spanish. This thesis will focus on processing English and Chinese. The ACE framework offers several benefits: a relatively rich set of tasks; substantial training data for these tasks; and evaluation scores which can be used as a benchmark for the experimental results.

[2] http://www.nist.gov/speech/tests/ace/ace07/et/index.htm

3. PRIOR WORK

The work presented in this thesis extends a substantial body of previous work.

3.1 Using Wider Context and Deeper Knowledge for IE

Some previous NLP systems have attempted to apply wider context and deeper analysis. The following sections present some examples for improving name tagging and coreference resolution.

3.1.1 Using Wider Context for Name tagging

A wide variety of unified learning algorithms have been applied to the name tagging task, including HMMs (Bikel et al., 1997), maximum entropy models (Borthwick, 1999; Chieu and Ng 2002; Florian et al., 2007), Decision Trees (Sekine et al., 1998), Conditional Random Fields (McCallum and Li, 2003), Class-based Language Model (Sun et al., 2002), Agent-based Approach (Ye et al., 2002) and Support Vector Machines (Takeuchi and Collier, 2002).

However, the performance of these models has been limited by the amount of labeled training data available to them and the range of features which they employ. In particular, people have spent considerable effort in engineering appropriate features to classify an instance of a name based on the information about that instance alone; most of these involve internal name structure and the immediate local context of that instance – typically, one or two words preceding and following the name. If a name has not been seen before, and appears in a relatively uninformative context, it becomes very hard to classify.

Some other name tagging systems have explored global information for name tagging. Some approaches have incorporated a name cache or similar mechanism, in which tokens or complete names which have been previously assigned a tag are available as features in tagging the remainder of a document.

For example, a voted cache model takes into account the number of times a particular name has been assigned each type of tag. (Borthwick, 1999) made a second tagging pass which uses information on token sequences tagged in the first pass. Chieu and Ng (2002) and Florian et al. (2004) report on name taggers which use as features the contexts of other instances of the same token in a document – an indirect and somewhat convoluted (but, apparently, effective) way of using coreferring mentions.

Finkel et al. (2005) used Gibbs sampling, a method to perform approximate inference in factored probabilistic models, to incorporate global knowledge for entity extraction. They achieved an error reduction of up to 9% over two baseline systems.

3.1.2 Using Deeper Knowledge for Coreference Resolution

Coreference resolution was a prime topic in the earlier studies of integrated, deep semantic systems. Much of the earlier work in coreference resolution (from the 1970's and 1980's, in particular) relied heavily on deep semantic analysis and inference procedures (Charniak 1972; Charniak 1973; Wilensky 1983; Carbonell and Brown 1988; Hobbs et al. 1993). Using these methods, researchers were able to give accounts of some difficult examples, often by encoding quite elaborate world knowledge. Unfortunately, it proved very hard to scale up such accounts to handle a broad range of examples. Capturing sufficient knowledge to provide adequate coverage of even a limited but realistic domain was very difficult. Applying these coreference resolution methods to a broad domain would require a large scale knowledge-engineering effort.

The focus for the last decade has been primarily on broad coverage systems using relatively shallow knowledge, and in particular on corpus-trained statistical models. Most of these coreference resolution systems use representations built out of the lexical and syntactic attributes of the mentions for which reference is to be established. These attributes may involve string matching, agreement, syntactic distance, and positional information, and they tend to rely primarily on the immediate context of the noun phrases (with the possible exception of sentence-spanning distance measures such as Hobbs distance). Though gains have been made with such methods (Tetreault, 2001; Mitkov, 2001; Soon et al., 2001; Ng and Cardie, 2002; Ng, 2005; Yang et al., 2003; Yang et al., 2006; Luo et al., 2004; Luo and Zitouni, 2005), there are clearly cases where this sort of local information will not be sufficient to resolve coreference correctly.

Some of of these systems attempt to apply shallow semantic information. (Ge et al. 1998) incorporate gender, number, and animaticity information into a statistical model for coreference resolution by gathering coreference statistics on particular nominal-pronoun pairs. (Soon et al., 2001) use WordNet to test the semantic compatibility of individual mention pairs. In general these approaches do not explore the possibility of exploiting the global semantic context provided by the document as a whole. Vieira and Poesio (2000), Harabagiu et al. (2001), and Markert and Nissim (2005) explore the use of WordNet for different coreference resolution subtasks. All of them present systems which infer coreference relations from a set of potential antecedents by means of a WordNet search. Tetreault and Allen (2004) use a semantic parser to add semantic constraints to the syntactic and agreement constraints in their Left-Right Centering algorithm. Bean and Riloff (2004) have sought to acquire automatically some semantic patterns that can be used as contextual information to improve coreference resolution, using techniques adapted from information extraction. Their experiments were conducted on collections of texts in two topic areas (terrorism and natural disasters).

3.2 Multiple Hypothesis Propagation and Ranking Algorithms

Traditional statistical NLP methods have generated a single hypothesis as their output. Generating N-Best hypotheses in speech recognition is a well-studied problem. One of the early works is the BBN model introduced in (Chow and Schwartz, 1989) for speech recognition. (Eppstein, 2001) listed an extensive bibliography. (Mohri and Riley, 2002) proposed a very efficient algorithm to find N-Best hypotheses.

In recent years, re-ranking techniques have been successfully applied to enhance the performance of NLP components based on generative models. A baseline generative model produces N-best hypotheses, which are then re-ranked using a rich set of local and global features in order to select the best hypothesis.

3.2.1 Algorithms

There has been a considerable body of work in the last few years on various trainable ranking algorithms such as Coordinate Descent RankBoost (Rudin et al., 2005) and PRank (Crammer and Singer, 2001).

Three state-of-the-art supervised ranking techniques are adopted in this thesis, Maximum Entropy Modeling-based Ranking (MaxEnt-Rank), Support Vector Machine-based Ranking (SVMRank) (Joachims 2002; Herbrich et al., 2000), and a new boosting-style ranking approach called p-Norm Push Ranking (Rudin, 2006). This algorithm is a generalization of RankBoost (Freund et al. 1998; Freund et al. 2003) which concentrates specifically on the top portion of a ranked list. Our work in this thesis was the first successful application for this approach.

3.2.2 Applications

These algorithms in the above section have been used primarily within the context of a single NLP component, with the most intensive study devoted to substantial improvements in name tagging, parsing and machine translation. Some prior work will be presented as follows.

3.2.2.1 Name Tagging

Collins (2002) applied RankBoost and Voted Perceptron (Freund and Schapire, 1999) to re-rank the 20 Best hypotheses generated from a maximum-entropy name tagger. The boosting algorithm is a modification of the method in (Freund et al., 1998), an adaptation of AdaBoost. These two methods achieved 15.6% to 17.7% relative reduction in error rates over the baseline.

Zhai et al. (2004) applied the mechanism of weighted voting among multiple hypotheses to re-rank Chinese name hypotheses for speech data. This voting scheme is incorporated as one feature in the name re-ranker.

3.2.2.2 Machine Translation

Re-ranking techniques have also been applied to SMT, such as MaxEnt-Rank (Och and Ney, 2002) and gradient methods (Och, 2003). (Shen et al., 2004) applied a Voted Perceptron algorithm to MT Re-Ranking, and the resulting algorithm provided state-of-the-art performance in the NIST 2003 Chinese-English large data track evaluation.

3.2.2.3 Parsing

In recent years, re-ranking techniques have resulted in significant improvements in parsing. (Collins and Duffy, 2002) applied the Voted Perceptron algorithm to re-ranking parsing results and obtained a 5.1% relative reduction in error rates. Other various machine learning algorithms have been employed in parse re-ranking, such as RankBoost (Collins, 2003; Collins and Koo, 2003; Kudo et al., 2005, Chen et al., 2002) and SVMRank (Shen and Joshi, 2003). These techniques have resulted in a 13.5% error reduction in labeled recall/precision over the previous best generative parsing models.

Shen and Joshi (2003) compared two different sample creation methods, and presented an efficient training method by separating the training samples into subsets. Shen and Joshi (2004) introduced a new perceptron-like ordinal regression algorithm for parse re-ranking.

MaxEnt-Rank (Charniak and Johnson, 2005) and Kernel Based Ranking Methods (Henderson and Titov, 2005) have also proved effective to enhance the performance of state-of-the-art parsers.

3.2.2.4 Semantic Role Labeling

Recent work on Semantic Role Labeling has shown that to achieve high accuracy a joint inference on the predicate argument structures of the entire tree should be applied. Toutanova (2005) used log-linear re-ranking model based on constraints among arguments, and obtained competitive performance in CONLL 2005. Moschitti et al.(2006) applied tree kernels to re-rank the candidate arguments for each predicate.

3.2.2.5 Spoken Language Processing

Re-ranking techniques are particularly effective for more noisy data such as speech transcripts. Huang et al. (2007) applied RankBoost algorithm to enhance Chinese part-of-speech tagging, and achieved 18% relative reduction in errors compared to the baseline tagger without re-ranking.

3.3 Interactions among NLP Subtasks

This section will present some previous work on capturing interactions among NLP subtasks. The need for interaction of different natural language components has been recognized at least since the 1960’s. To see how these interactions might be structured, a number of integrated systems were built for microworlds (Winograd, 1972), children’s stories, and other toy applications. Studies were also conducted of formal (logical) representations for integrating this information, such as the work of Hobbs et al. (1993). There were also extensive studies of the role of text coherence in language understanding (Hobbs, 1985; Halliday and Hasan, 1976).

3.3.1 Interactions among Entity Type, Entity SubType and Mention Type Tagging

(Florian et al., 2006) adopted a joint model to incorporate the mutual dependencies among entity type, entity subtype and mention type tagging for the mention extraction task, and achieved significant improvement in F-measure over the monolithic model ("All-In-One" model) for English, Chinese and Arabic.

3.3.2 Interactions between Name Structure Parsing and Coreference Resolution

(Charniak, 2001) employed a simple probabilistic name model, with a linear sequence of components. Charniak has demonstrated how coreference resolution can improve the learning of name structure, and name structure can improve coreference resolution; the system got 97% correct English name structures.

3.3.3 Interaction between Mention Detection and Coreference Resolution

Joint Inference between mention detection and coreference resolution has become a topic of keen interest. Wellner et al. (2004) used a conditionally-trained graphical model to incorporate the interaction between extraction of mentions (here, mentions mean various database fields such as title, author, journal, year, etc.) and coreference resolution. N-Best mention hypotheses were generated and then the coreference was accomplished by approximate inference via a greedy graph partitioning algorithm. Experimental results showed significant improvements in coreference by using uncertainty information from extraction, and in extraction accuracy using results of coreference.

3.3.4 Interaction between Mention Detection and Relation Detection

There have been recent efforts to simultaneously extract mentions and relations by capturing their mutual dependencies and exploiting the global inference based on the interactions.

Roth and Yih (2002, 2004, 2007) combined information from named entities and semantic relation tagging, adopting a similar overall goal as in this thesis but using a quite different approach based on linear programming. The predictors that identify entities and relations among them are first learned from local information in the sentence. The constraints induced by the mutual dependencies among entity types and relations constitute a relational structure over the outcomes of the predictors and are used to make global inference.

Roth and Yih (2002) formulated global inference using a Bayesian network, where they captured the influence between a relation and a pair of entities via the conditional probability of a relation, given a pair of entities. This approach however, could not exploit dependencies between relations.

Roth and Yi (2004; 2007) proposed a Linear Programming (LP) formulation in which to cast inference. Given name boundaries in the text, separate classifiers are first trained for name classification and semantic relation detection. Then, the output of the classifiers is used as a conditional distribution given the observed data. This information, along with the constraints among the relations and entities (specific relations require specific classes of names), is used to make global inferences by linear programming for the most probable assignment. They obtained significant improvements in both name classification and relation detection. Different methods, defined combining in different ways the entity and relation classifiers, were evaluated: In the first one, a basic entity classifier, identical to the entity classifier in the separate approach, is trained. Its predictions on the two entity arguments of a relation are then used conjunctively as additional features in training the relation classifier. Similarly, a second pipeline first trains the relation classifier; its output is then used as additional features in the entity classifier. The third pipeline model is the combination of the above two. A final step tests the conceptual upper bound of the entity/relation classification problem. It assumes that the entity classifier knows the correct relation labels and the relation classifier knows the right entity labels. This additional information is then used as features in training and testing.

Roth and Yih limited themselves to name classification, assuming the identification (the exact boundaries of entities) given. This may be a natural subtask for mixed case English data, where capitalization is a strong indicator of a name so name identification is relatively easy. But this assumption may not be available in practice, thus the approach is much less useful for other languages such as Chinese, where there is no capitalization or word segmentation, and boundary errors on name identification are frequent. Expanding their approach to cover identification would have greatly increased the number of hypotheses and made their approach slower.

Choi et al. (2006) applied this approach for identifying opinion expressions and the relations between opinions. They showed that such global, constrait-based inference can significantly boost the performance of both opinion identification and relation detection.

3.3.5 Interaction between Information Extraction and Semantic Role Labeling

A few previous works have tried to bridge IE and other NLP tasks such as semantic role labeling (SRL). Some shallow IE analysis such as name tagging has been used to pre-process the input texts for SRL systems (Carreras and Marquez, 2004; Carreras and Marquez, 2005). For example, a person mention (name or nominal phrase) is likely to appear as an "Agent" argument for the predicate "announce"; a location mention is likely to act as an "ARGM-LOC" argument.

On the other hand, recently SRL has been applied to enhance some IE components. Wattarujeekrit (2005) used semantic roles to enhance name tagging in the molecular biology domain. A domain-customizable IE system may be designed if we know: (a) predicates ("triggers") relevant to a domain; and (b) which of their arguments fill template slots. So it's natural to apply a semantic role labeling (SRL) system that can generate predicate-argument structures on the output of full parsers to improve IE. One of such attempts was the work described in (Surdeanu et al., 2003). They obtained about 14% better IE F-measure using SRL results. (Grishman et al., 2005) achieved significant improvement in event detection using the patterns based on predicate-argument structures (Meyers et al., 2001) connecting the trigger to all the event arguments.

3.3.6 Interaction between Entity Extraction and Entity Translation

All the stage interactions described above focused on the monolingual analysis pipeline. (Huang and Vogel, 2002) presented a cross-lingual joint inference example to improve the extracted named entity translation dictionary and the entity annotation in a bilingual training corpus. They used a more ‘traditional’ approach to encode the interaction between different NLP tasks: sharing the training resources across tasks. This thesis expands the idea of alignment consistency to the task of entity extraction in a monolingual test corpus.

(Florian et al., 2007) used a similar idea to expand the mention detection training data for Spanish by translating and projecting an annotated English corpus, and significantly improved Spanish mention detection. This approach is expected to be effective in particular when machine translation performs well for the given language pair.

3.4 Conclusion

All these works noted the advantage of exploiting multiple hypotheses and then using richer knowledge for re-ranking, but the features in re-ranking are all restricted to the original goal task itself. The re-ranking models for name tagging, for example, normally rely on structures generated within the baseline name tagger only. (Collins, 2002) was limited to local features involving lexical information and 'word-shape' in a 5-token window. In contrast, the name re-ranker (secion 7.1) in this thesis will make use of a richer set of global features, involving the detailed evidence from the subsequent IE stages such as coreference resolution. In this way the re-ranker can incorporate information from a wider context, spanning the entire document and even going across documents.

In contrast to the traditional approaches of encoding sophisticated semantic features to enhance NLP tasks, this thesis will present an approach of using semantic relation detection results to infer and correct coreference results. Relation detection implicitly selects relevant deeper context. This allows us to naturally capture deep, although still relatively lightweight, semantic knowledge with low cost.

In addition, compared to the prior work using stage interactions, this thesis will focus more on capturing selectional preferences (e.g. probabilities of semantic types as arguments of particular semantic relations as computed from the corpus) instead of boolean constraints as shown in the linear programming framework by Roth and Yih (2002, 2004, 2007).

4. A BASELINE INFORMATION EXTRACTION AND TRANSLATION SYSTEM

This thesis has a solid base from which to work, in the form of Chinese and English IE systems which have been developed and applied over the course of the last several ACE evaluations (2002-2007). The core stages include entity detection and tracking, relation detection, event pattern acquistion and entity translation.

4.1 Baseline System Overview

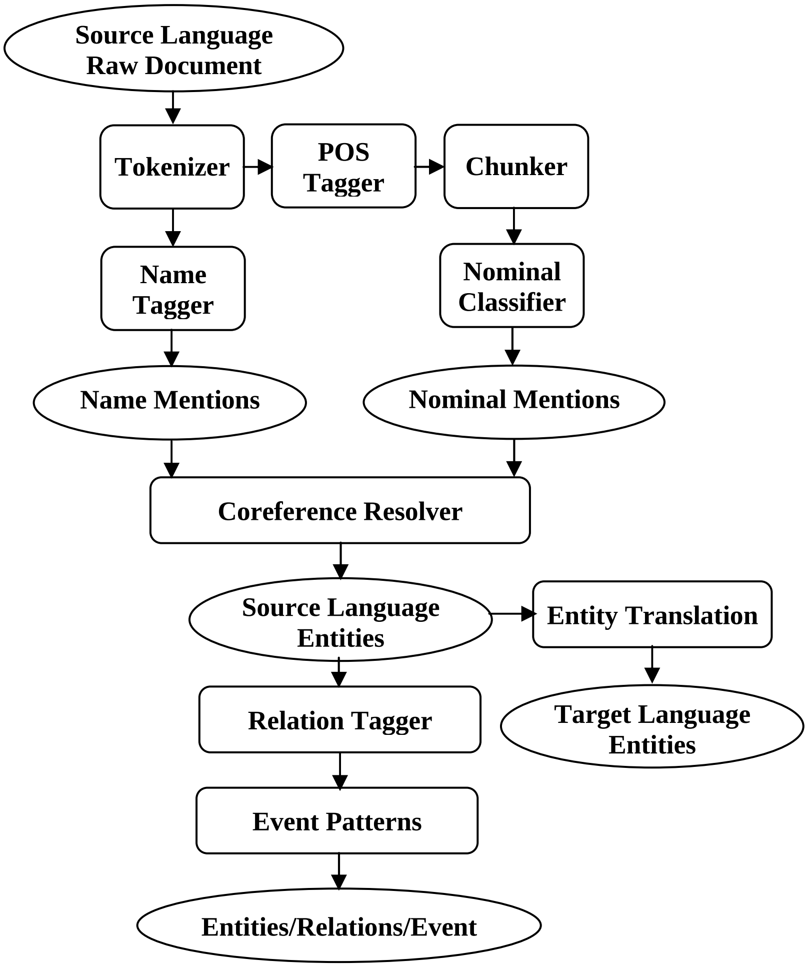


Figure 4-1. Baseline Information Extraction and Entity Translation Pipeline

The overall architecture of our baseline English and Chinese IE pipeline is presented in Figure 4-1. In addition, we developed a Chinese-to-English entity translation component.

4.2 Baseline System Components

Each component is briefly described in the following subsections.

4.2.1 Tokenizer

At the first step the input text is divided into sentences and tokenized. The Chinese system uses a word segmenter from Tsinghua University similar to the version described in (Wan and Luo, 2003). For English the tokens are looked up in a large general English dictionary that provides part-of-speech information and the base form of inflected words.

4.2.2 Name Tagger and Name Structure Parsing

The Chinese baseline name tagger consists of a HMM tagger augmented with a set of post-processing rules. The HMM tagger generally follows the Nymble model (Bikel et al, 1997). Within each of the name class states, a statistical bigram model is employed, with the usual one-word-per-state emission. The various probabilities involve word co-occurrence, word features, and class probabilities. Since these probabilities are estimated based on observations seen in a corpus, several levels of "back-off models" are used to reflect the strength of support for a given statistic, including a back-off from words to word features, as for the Nymble system.

The following improvements have been made to the model. To take advantages of Chinese language-specific name structures, name structure parsing is done using an extended HMM which includes a larger number of states (14). The name structure parsing results include the family name and given name of persons, the prefixes of the diminuitive names, the modifiers and suffixes of organization names. This new HMM can handle name prefixes and suffixes, and transliterated foreign names separately. The event trigger word lists and a title list were extracted from the ACE05 event training corpus, and a TIME word list was extracted from TIMEX data. These were then used to construct additional features in the Nymble model. In total 19 features are employed in this baseline tagger.

Finally a set of post-processing heuristic rules are applied to correct some omissions and systematic errors using name lists (for example, a list of all Chinese last names; lists of organization and location suffixes) and particular contextual patterns (for example, verbs occurring with people's names). They also deal with abbreviations and nested organization names. List matching is applied to identify facility, weapon and vehicle names.

The English name tagger is based on a HMM including six states for each of the five main name types (Person, GPE, Location, Location, and Facility), as well as a not-a-name state. These six states correspond to the token preceding the name; the single name token (for names with only one token); the first token of the name; an internal token of the name (neither first nor last); the last token of the name; and the token following the name. These multiple states allow the HMM to capture context information and limited information about the internal structure of the name.

4.2.3 POS Tagger and Chunker

For POS tagging, Chinese adopts the Tsinghua Chinese part-of-speech (POS) tagger and while the English POS tagger is based on a bigram HMM . The Chinese chunker is based on SVM and English chunker based on MaxEnt. The main features used in chunking are the bigram conjunctions of POS features. For example, for a word W_i with POS tag P_i, assume the POS tags for the context of W_i are P_{i-2}, P_{i-1}, P_{i+1}, $P_{i+2,}$ we use the conjunction of these tags as features: $P_{i-2}P_{i-1}$, $P_{i-1}P_i$, P_iP_{i+1} and $P_{i+1}P_{i+2}$.

4.2.4 Nominal Mention Tagger

Entity type assignment for the nominal heads is done by table look-up. The nominal head lists are generated from ACE training corpora, and a small part from Chinese Hownet[3].

4.2.5 Coreference Resolver

The baseline coreference resolver is a two-stage process. First, high-precision heuristic rules make some positive and negative reference decisions. Rules include simple string matching (e.g., names that match exactly are resolved), agreement constraints (e.g., a nominal will never be resolved with an entity that doesn't agree in number), and reliable syntactic cues (e.g., mentions in apposition are resolved). When such a rule applies, it assigns a confidence value of 1 or 0 to a candidate mention-antecedent pair.

The remaining pairs are assigned probability values by a collection of maximum entropy models. To address the special properties of different mention types, the system is separated into different models for names, nominals, and pronouns. Each model

[3] www.keenage.com/

operates on a distinct feature set, and for each instance only one of these three models is used to produce a probability that the instance represents a correct resolution of the mention. A threshold is applied to this probability: if some resolution has a confidence above the threshold, the highest confidence resolution will be made. Otherwise the mention is assumed to be the first mention of an entity.

Both the English and the Chinese coreference models incorporate the following main features:

- ❑ representing agreement of various kinds between mentions (number, gender, humanness)
- ❑ degree of string similarity
- ❑ synonymy between mention heads
- ❑ measures of distance between mentions (such as the number of intervening sentences)
- ❑ the presence or absence of determiners or quantifiers

For each pair of anaphor and its candidate antecedent *(i, j)*, the features are shown in Table 4-1. For English name re-ranker, a separate rule-based English coreference resolution system (Grishman, 2004) was used.

Language	Feature Names	Feature Descriptions
English & Chinese	STR_MATCH	substring match
	Detj	if j starts with “the”, “this”, “that”, “these”, “those”…
	NUMBER	if noun is a singular, plural or unknown
	ALIAS	if i is a person alias name to j
	Apposition	if j is in apposition to i
	Synonymy	if i is the synonym of j
	Identical	if i and j are identical
	Positional	distance between the mentions in terms of # of sentences
	SameHead	if i and j have the same heads
	Human	in PER pair, if the nominal mention has Human feature (Chinese determines from Hownet; English checks pronoun)
English Only	AnaphorHead	head of j
	EntityType	the entity types of I and j
	MentionType	the mention type of I and j
	Possessive	if i and j are possessive
	Generic	if j is generic mention
	ModifierMatch	if i and j have compatible left/right modifiers
	Quantifier	Quantifier attached to j, if any
	HobbsDistance	Hobbs distance between i and j
Chinese Only	DIST	if (i, j) are in the same sentence
	GPEAbb	if i is an abbreviation name of j
	GPEsuffix	if nominal has the same GPE suffix word as name
	Beginning	if i locates in the beginning of sentence

Table 4-1. Features for the Baseline Coreference Resolver

4.2.6 Relation Tagger

The relation tagger uses a K-nearest-neighbor algorithm, in which the training examples are recorded and during test the training example most similar to the test example is used to classify the instance. (In case of multiple training examples equally similar to a test case, frequency in the training corpus is used to break ties.) A mention pair is considered as a possible instance of a relation only when:

- the number of mentions between their heads is less than a threshold (different threshold values for different types)

- the coreference probability produced for the pair by the baseline resolver is lower than a threshold

Each training / test example consists of the pair of mentions and the sequence of intervening words. Associated with each training example is either one of the ACE relation types or no relation at all. A distance metric between two examples is defined based on:

- whether the heads of the mentions match
- whether the ACE types of the mentions match (for example, both are people or both are organizations)
- whether the ACE subtypes of the mentions match
- whether the sequence of the heads of the constituents between the two mentions match (the English system basically follows the highest cut through the parse between the two mentions, Chinese system prunes the intervening words by chunking information and stop-word list)
- in the English system, whether the syntactic relation paths between the two mentions, as obtained from a parse tree match (Grishman et al., 2005).

To tag a test example, the algorithm finds the k nearest training examples (where k = 3) and use the distance to weight each neighbor, then select the most common class in the weighted neighbor set.

To provide a crude measure of the confidence of the relation tagger, two thresholds are defined, D_{near} and D_{far}.

If the average distance d to the nearest neighbors $d < D_{near}$, it is considered as a *definite* relation;

If D_{near} < d < D_{far}, it is considered as a *possible* relation;

If d > D_{far}, the tagger assumes that no relation exists (regardless of the class of the nearest neighbor).

4.2.7 Event Pattern Acquisition

Chinese event patterns are extracted from the ACE05 training corpus, For each event instance the trigger word (A trigger is the word that most clearly expresses the event occurrence) is replaced by its event type and subtype, and each argument by its entity type. Then POS tagging and chunking are applied on each event training instance. Then the patterns are edited and generalized by hand, replacing tokens by their POS tag, chunk type, or a wild card or deleting them entirely if they are not relevant to detecting the event type. Some patterns are collapsed, and some patterns which appear too specific or too general are deleted. To insure that patterns are not over-generalized by the hand editing, the training corpus is split in two and patterns derived from one half are, after hand editing, applied to the other half to review their accuracy in event prediction. Additional event trigger words that appear frequently in name contexts are also collected from a syntactic dictionary, a synonym dictionary and Chinese PropBank V1.0 (Xue and Palmer, 2003). In the test procedure, each document is annotated with POS tagging and chunking, and then scanned against the patterns derived from the training corpus.

The experiments in this thesis don’t use English event tagger directly. Instead the co-occurrence information between name type and verbs is extracted from COMLEX (Macleod et al., 1998). We will present the details in section 7.1.2.6.

4.2.8 Entity Translation

The entities generated by Chinese IE are then translated into English. The RWTH Aachen Chinese-to-English machine translation system (Zens and Ney, 2004) is used. It's a statistical, phrase-based machine translation system which memorizes all phrasal translations that have been observed in the training corpus. It computes the best translation using a weighted log-linear combination of various statistical models: an n-gram language model, a phrase translation model and a word-based lexicon model. The latter two models are used in source-to-target and target-to-source directions. Additionally, it uses a word penalty and a phrase penalty.

The model scaling factors are optimized on the development corpus with respect to the BLEU score similar to (Och 2003). Almost all bilingual corpora provided by LDC were used for training, which account for about 200 million running words in each language. Language modeling used the English part of the bilingual training corpus and in addition some parts of the English GigaWord corpus. The total language model training data consists of about 600 million running words. This MT system produces a translation for each source document, and also the word-to-word mapping derived from phrase-based alignment.

For each individual Chinese mention, this MT system is used to translate it in isolation. Here, the only difference from text translation is that, as subsentential units are translated, sentence boundaries are not assumed at the beginning and end of each input segment. This isolated translation scheme is applied instead of the entity projection based on word alignment because:

- It produces less alignment noise. Note the word alignments are indirectly derived from phrase alignment, and thus context words often become noise for mention translation
- Manual evaluation on a small development set showed that isolated translation obtains (about 14%) better F-measure.

4.3 Conclusion

The above framework represents a monolingual English IE system, a Chinese IE system, and a Chinese to English entity translation system. These are relatively simple, cleanly structured components. All of the enhancements presented in this thesis will be evaluated using these components as a baseline.

5. TYPES OF LINGUISTIC INTERACTION

Our focus for this chapter is on exploring specific interactions to enhance the performance of information extraction. They are presented in some detail, both as evidence of the promise of our approach and as an indication of how these can be captured and organized into a re-ranking model in the next chapter.

According to the different range of knowledge sources we explored, we divided the interactions into two types:

(1) Mono-lingual Interaction: within IE pipeline, using the feedback from subsequent stages to correct the results of the previous stages;

(2) Cross-lingual Interaction: using the feedback from machine translation to improve IE.

In the following these global interactions will be presented based on their motivation from linguistic intuitions and the error analysis from the baseline systems.

5.1 Cross-Stage Interaction

This section describes the interactions which improve name tagging (section 5.1.1) and coreference resolution (section 5.1.2) by using the feedback from subsequent IE stages.

5.1.1 Stage Interactions for Correcting Name Tagging Errors

As our first experiment, we investigate how name tagging can be improved using later IE stages.

A detailed understanding of name tagging errors is a prerequisite for further performance improvements. The task of name tagging can be decomposed into two subtasks:

- Name Identification – The process of identifying name boundaries in the sentence.
- Name Classification – Given the correct name boundaries, assigning the appropriate name types to them.

Assuming:

S: name set returned by the name tagger

K: key name set

then the name tagging errors can be further subdivided by the following types:

Spurious Error = {s | s∈ S, k∈ K, s doesn't overlap with any k}.;

Missing Error = {k | s∈ S, k∈ K, k doesn't overlap with any s};

Boundary Error = {s | s∈ S, k∈ K, s partially overlaps with k}.

Classification Error = {s | s*∈S, k∈K, s and k match on boundaries but have different name types}.*

The next sections shall illustrate, through a series of examples, the potential for feedback from subsequent IE stages to correct different types of name tagging errors in English and Chinese texts.

5.1.1.1 Name Identification and Classification Error Analysis

In mixed-case English texts, most proper names are capitalized. So capitalization provides a crucial clue for name boundaries.

In contrast, a Chinese sentence is composed of a string of characters without any word boundaries or capitalization. Even after word segmentation there are still no obvious clues for the name boundaries. However, the following coarse “usable-character” restrictions can be applied to reduce the search space.

Standard Chinese family names are generally single characters drawn from a set of 437 family names (there are also 9 two-character family names, although they are quite infrequent) and given names can be one or two characters (Gao et al., 2005). Transliterated Chinese person names usually consist of characters in three relatively fixed character lists (Begin character list, Middle character list and End character list). Person abbreviation names and names including title words match a few patterns. The suffix words (if there are any) of Organization and GPE names belong to relatively distinguishable fixed lists. However, this “usable-character” restriction is not as reliable as the capitalization information for English, since each of these special characters can also be part of common words.

English and Chinese HMM name taggers (as described in section 4.2.2) are applied to identify names, and use best-first search to generate N-Best multiple hypotheses for each sentence, and also compute the *margin* metric defined as follows:

Margin = LogProbability (Best Hypothesis) – LogProbability (Second Best Hypothesis)

Figure 5-1 shows the identification F-Measure for the baseline (the first hypothesis), and the N-best upper bound, the best of the N hypotheses tested on 100 ACE Chinese texts (N=30) and 20 English texts (N=20), scored with respect to the official ACE04 keys prepared by the Linguistic Data Consortium. The results are scored using different models: English MonoCase (EN-Mono, without capitalization), English Mixed Case

(EN-Mix, with capitalization), Chinese without the usable character restriction (CH-NoRes) and Chinese with the usable character restriction (CH-WithRes).

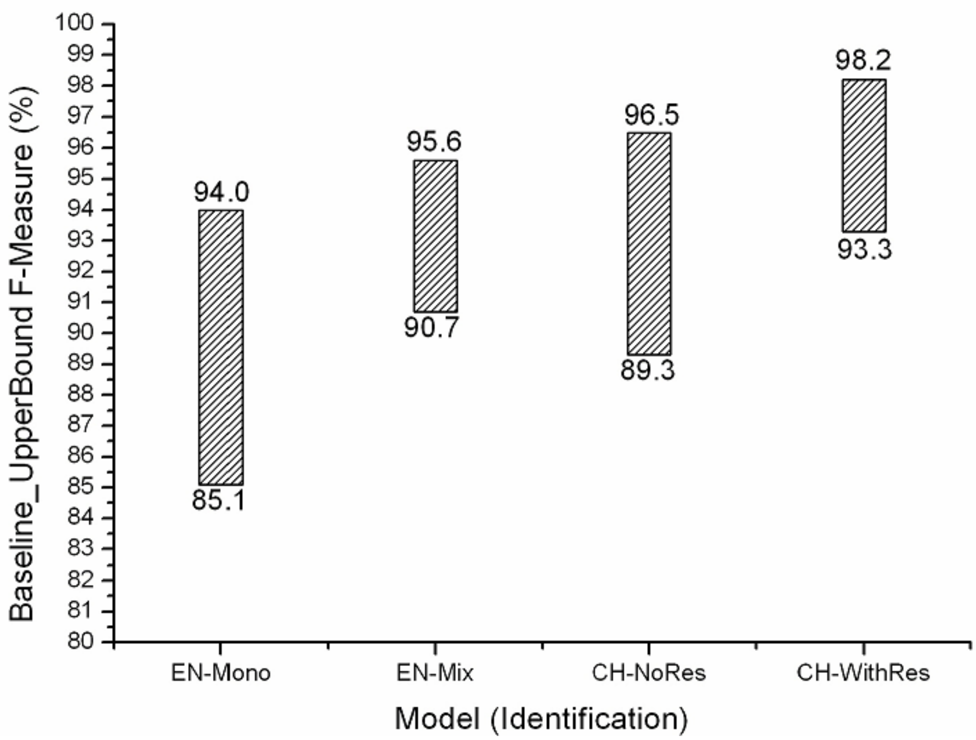


Figure 5-1. Baseline and Upper Bound of Name Identification

Figure 5-1 shows that capitalization is a crucial clue in English name identification (improving the F measure by 5.6% over the mono-case score). The “usable” character restriction plays a major role in Chinese name identification, increasing the F-measure 4%. The figure also shows that the best of the top N hypotheses is very good.

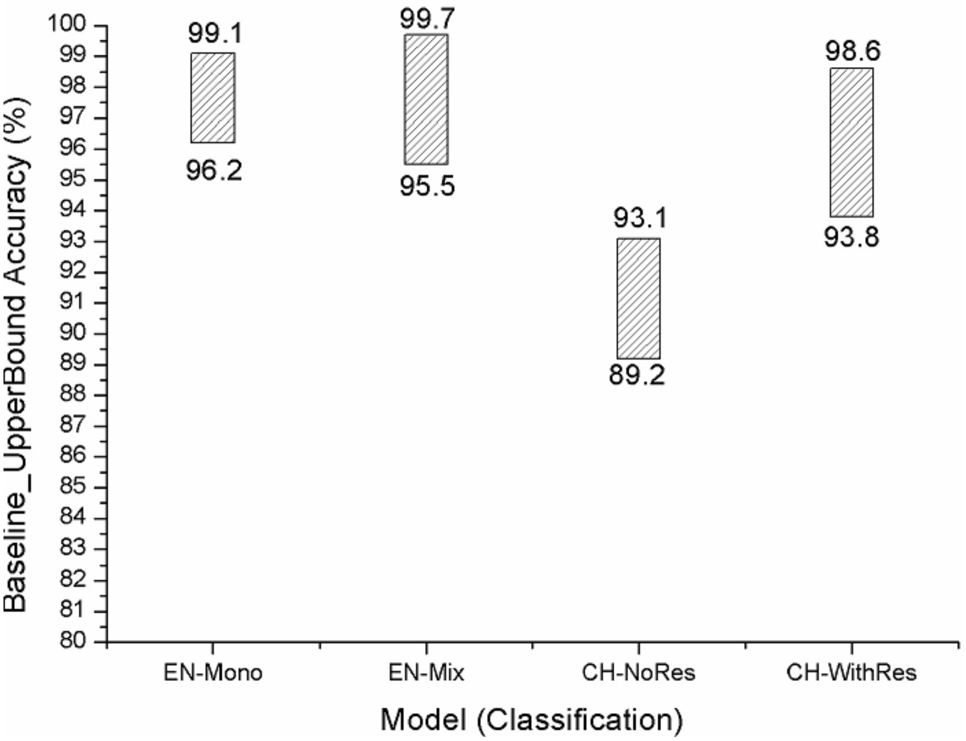


Figure 5-2. Baseline and Upper Bound of Name Classification

Figure 5-2 shows the classification accuracy of the above four models. The figure indicates that capitalization does not help English name classification.

The Chinese name identification errors can be divided into missed names (21%), spurious names (29%), and boundary errors (50%). Confusion between names and nominals (phrases headed by a common noun) is a major source of both missed and spurious names (56% of missed, 24% of spurious). In a language without capitalization, this is a hard task even for people; one must rely largely on world knowledge to decide whether a phrase (such as the "criminal-processing team") is an organization name or merely a description of an organization. The other major source of missed names is words not seen in the training data, generally representing minor cities or other locations in China (28%). For spurious names, the largest source of error is names of a type not included in the key (44%) which are mistakenly tagged as one of the known name types. The following sections will show that different types of knowledge are required for correcting different types of errors.

5.1.1.2 Name Tagging from an IE View

From Chapter 4 we can see that several stages follow name tagging in the IE pipeline such as coreference, semantic relation extraction and event extraction. All these stages are performed after name tagging since they take names as input "objects". However, the feedback from these subsequent stages can also provide valuable constraints to identify and classify names. Each of these stages connects the name candidate to other linguistic elements in the sentence, document, or corpus, as shown in Figure 5-3.

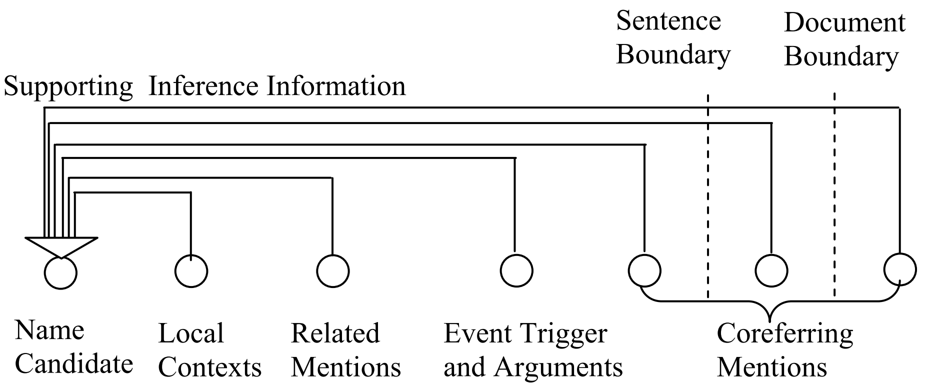


Figure 5-3. Name Candidate and Its Global context

Specifically, this thesis will take advantage (among other properties) of the coherence of a discourse (Hobbs, 1985; Halliday and Hasan, 1976): that a correct analysis of a discourse reveals a large number of connections between its elements, and so (in general) a more tightly connected analysis is more likely to be correct.

Consider, for example, the problem of identifying person names; the name tagger, in isolation, may have difficulty deciding whether a sequence of tokens is a person name, a name of another type, or not a name at all. If the name (or a similar name) appears elsewhere in the document, coreference resolution may help resolve the ambiguity; if an event tagger determines that the name appears as the employee of some organization or the attacker of a bombing event, this event information can help resolve the ambiguity. Therefore a correct name tagging (for example) will license more coreference relations as well as more semantic relations or events (such as 'X is located in Y', 'X works for Y', 'X attacked Y', etc.). The baseline name tagger (HMM) uses very local information; such feedback from later extraction stages allows us to draw from a wider context in making final name tagging decisions.

In the following two related Chinese (translated) texts are used as examples, to give some intuition of how these different types of linguistic evidence improve name tagging.[4]

Example 5-1: Yugoslav election

> […] More than 300,000 people rushed the *<bei er ge le>*$_0$ congress building, forcing *<yugoslav>*$_1$ president *<mi lo se vi c>*$_2$ to admit frankly that in the Sept. 24 election he was beaten by his opponent *<ke shi tu ni cha>*$_3$.
>
> *<mi lo se vi c>*$_4$ was forced to flee *<bei er ge le>*$_5$; the winning opposition party's *<sai er wei ya>*$_6$ *<anti-democracy committee>*$_7$ on the morning of the 6th formed a *<crisis-handling committee>*$_8$, to deal with transfer-of-power issues. This crisis committee includes police, supply, economics and other important departments.

[4] Rather than offer the most fluent translation, we have provided one that more closely corresponds to the Chinese text in order to more clearly illustrate the linguistic issues. Transliterated names are rendered phonetically, character by character.

> In such a crisis, people cannot think through this question: has the *<yugoslav>*$_{9}$ president *<mi lo se vi c>*$_{10}$ used up his skills?
>
> According to the official voting results in the first round of elections, *<mi lo se vi c>*$_{11}$ was beaten by *<18 party opposition committee>*$_{12}$ candidate *<ke shi tu ni cha>*$_{13}$. […]

Example 5-2: Biography of these two leaders

> […]*<ke shi tu ni cha>*$_{14}$ used to pursue an academic career, until 1974, when due to his opposition position he was fired by *<bei er ge le>*$_{15}$ *<law school>*$_{16}$ and left the academic community.
>
> *<ke shi tu ni cha>*$_{17}$ also at the beginning of the 1990s joined the opposition activity, and in 1992 founded *<sai er wei ya>*$_{18}$ *<opposition party>*$_{19}$.
>
> This famous new leader and his previous classmate at law school, namely his wife *<zuo li ka>*$_{20}$ live in an apartment in *<bei er ge le>*$_{21}$.
>
> The vanished *<mi lo se vi c>*$_{22}$ was born in *<sai er wei ya>*$_{23}$ ‘s central industrial city. […]

5.1.1.3 Interaction Between Name Tagging and Name Structure Parsing

Constraints and preferences on the structure of individual names can capture local information missed by the baseline name tagger. They can correct several types of identification errors, including in particular boundary errors. For example, "*<ke shi tu ni cha>*$_{3}$" is more likely to be correct than "*<shi tu ni cha>*$_{3}$" since "*shi*" (什) cannot be the first character of a transliterated name.

Name structures help classify names too. For example, "*anti-democracy committee*$_{7}$" is parsed as "[Org-Modifier anti-democracy] [Org-Suffix committee]", and the first character is not a person last name or the first character of a transliterated person name, so it is more likely to be an organization than a person name.

5.1.1.4 Interaction Between Name Tagging and Relation Detection

Any context which can provide selectional constraints or preferences for a name can be used to correct name *classification* errors. Both semantic relations and events carry selectional constraints and so can be used in this way.

Relations are good indicators of the types of their arguments. For instance, if the "*Personal-Social/Business*" relation ("*opponent*") between "*his*" and "*<ke shi tu ni cha>$_3$*" is correctly identified, it can help classify "*<ke shi tu ni cha>$_3$*" as a person name. Relation information is sometimes crucial to classifying names. "*<mi lo se vi c>$_{10}$*" and "*<ke shi tu ni cha>$_{13}$*" are likely person names because they are "*employees*" of "*<yugoslav>$_9$*" and "*<18 party opponent committee>$_{12}$*". "*<sai er wei ya>$_{23}$*"can be easily tagged as GPE because it has part-whole relation with "city".

Relation information can provide evidence for name identification too. The basic intuition is that a name which has been correctly identified is more likely to participate in a relation than one which has been erroneously identified. For a given range of margins from the HMM, Table 5-1 shows the probability that a Chinese name in the first hypothesis is correct, for names participating and not participating in a relation:

Margin	In Relation(%)	Not in Relation(%)
<4	90.7	55.3
<3	89.0	50.1
<2	86.9	42.2
<1.8	84.1	34.7
<1.5	81.3	28.9
<1.2	78.8	23.1
<1	75.7	19.0
<0.8	67.3	15.3
<0.5	66.5	14.3
<0.2	66.4	11.0

Table 5-1. Probability of a name being correct

Table 5-1 confirms that names participating in relations are much more likely to be correct than names that do not participate in relations. It also shows, not surprisingly, that these probabilities are strongly affected by the HMM margin. So it is natural to use participation in a relation (coupled with a margin value) as a valuable feature for name tagging.

5.1.1.5 Interaction Between Name Tagging and Event Detection

Information about expected sequences of constituents surrounding a name can be used to correct name *boundary errors*. In particular, event extraction is performed by matching patterns involving a "trigger word" (typically, the main verb or nominalization representing the event) and a set of arguments. When a name candidate is involved in an event, the trigger word and other arguments of the event can help determine the name boundaries. For example, in the sentence "*The vanished mi lo se vi c was born in sai er wei ya 's central industrial city*", "*mi lo se vi c*" is more likely to be a name than "*mi lo se*", "*sai er wei ya"* is more likely be a name than *"er wei"*, because these boundaries will allow us to match the event pattern "*[Adj] [PER-NAME] [Trigger word for 'born' event] in [GPE-NAME]'s [GPE-Nominal]*".

Events, like relations, can also provide effective selectional preferences to correctly classify names. For example, "<*mi lo se vi c*>$_{2,4,10,11,22}$" are likely person names because they are involved in the following events: "*claim*", "*escape*", "*built*", "*beat*", "*born*".

Besides these fine-grained ACE-type events, we incorporate all indicative verbs in the wide context to disambiguate name types. For example, in *"Chiao and McArthur clearly* *enjoyed their 240-mile-high construction work”*, *“Chiao”* can be confirmed as a person name because it appears as an argument of “enjoy”.

5.1.1.6 Interaction Between Name Tagging and Coreference Resolution

This section considers why coreference is expected to help name tagging. Unless a name is really well known (“George Bush”), it is likely to be referred to again, either by repeating the same name and/or describing it with nominal mentions in the text. Put another way, names which are recognized by the system but are not coreferenced with any other mentions are quite likely to be mistakes. These name mentions will have the same spelling (though if a name has several parts, some may be dropped) and same semantic type. So if the boundary or type of one mention can be determined with some confidence, coreference can be used to disambiguate other mentions, by favoring hypotheses which support more coreference.

For example, if “< *mi lo se vi c>*$_{2}$” is confirmed as a name, then “< *mi lo se vi c>*$_{10}$” is more likely to be a name than “< *mi lo se>*$_{10}$”, by refering to “< *mi lo se vi c>*$_{2}$”. Also “This crisis committee” supports the analysis of “<*crisis-handling committee>*$_{8}$” as an organization name in preference to the alternative name candidate “<*crisis-handling>*$_{8}$”.

For a name candidate, high-confidence information about the type of one mention can be used to determine the type of other mentions. For example, for the repeated person name “< *mi lo se vi c>*$_{2,4,10,11,22}$” type information based on the event context of one mention can be used to classify or confirm the type of the others. The person nominal “This famous new leader” confirms “<*ke shi tu ni cha>*$_{17}$” as a person name. “his wife” helps to classify <*zuo li ka>*$_{20}$ as a person name because they refer to the same entity.

To confirm this intuition, we gathered accuracy statistics on our Chinese baseline system output for names which are not on a list of high frequency names and are recognized by the HMM with a margin below some threshold. The results are shown in Table 5-2.

Number of Mentions/Entity	1	2	3	4	5	6	7	8	>8
PER	43.9	87.1	91.2	88.0	91.6	92.0	94.7	92.3	97.4
GPE	55.8	88.8	96.1	100	100	100	100	95.8	97.5
ORG	64.7	80.6	89.5	94.3	100	100	--	--	100

Table 5-2. Accuracy (%) of names with low margin

The table shows that the accuracy of name recognition increases as the entity includes more mentions. It also indicates that for singletons (names without coreferring mentions), the accuracy ranges from 43.9% for people names to 64.7% for organization names. For names with one coreferring mention, the accuracy improves to 80.6% for organizations and 87.1% for people; for those with more than one coreferring mention, the accuracies for all types are above 90%. So, although the singletons constitute only about 10% of all names, increasing their accuracy can significantly improve overall performance. Coreference information can play a great role here.

Take the 157 PER singletons as an example; 56% are incorrect names. Among the correct names, 71% can be confirmed by the presence of a title word or a Chinese last name. Therefore, without strong confirmation features, singletons are much less likely to be correct names. This feature particularly helps to disambiguate Chinese name abbreviations. Name abbreviations are difficult to recognize correctly due to a lack of training data. Usually people adopt a separate list of abbreviations or design separate rules (Sun et al. 2002) to identify them. But many wrong abbreviation names might be produced because they can appear as common words in text (e.g. “中” can mean “China” or “middle”; “以” can mean “Israel” or “as”). If one can check whether a link exists between a name abbreviation candidate and a full name, this information can be used to confirm or discard the candidate.

5.1.2 Stage Interactions for Correcting Coreference Errors

Coreference resolution has proven to be a major obstacle in building robust systems for information extraction, question answering, text summarization and a number of other natural language processing tasks. The lack of semantics in the current methods leads to a performance bottleneck. In order to correctly identify the discourse entities which are referred to in a text, it seems essential to reason over the semantic relations, as well as the event representations embedded in the text. In this thesis we explore whether coreference resolution can benefit from semantic knowledge sources from the detection of relations and events.

5.1.2.1 Interaction between Coreference Resolution and Relation Detection

Coreference is by definition a semantic relationship: two noun phrases corefer if they both refer to the same real-world entity. Therefore we expect a successful coreference system should exploit world knowledge, inference, and other forms of semantic relations in order to resolve hard cases. If, for example, two mentions refer to people who work for two different organizations (in the same time-frame), then these mentions are less likely to corefer. Further progress will likely be aided by flexible frameworks for representing and using the information provided by this kind of semantic relation between noun phrases.

The ACE relations provide a coarse classification of a broad range of predicates. Repeated relations may therefore provide a valuable additional source of information for coreference; these sources should be more reliable guides for coreference than simple lexical context or even tests for the semantic compatibility of heads and modifiers. Some examples from real texts below will support this intuition.

Example 5-3

> […] The ***lawyer*** for **Vice President Al Gore** said Wednesday their only chance for victory in his contest of the Florida election would be shattered if they have to wait until Saturday to begin counting disputed ballots, and they began an appeal to the Florida Supreme Court to do the counting itself, immediately.
>
> …
>
> And it could take weeks if not months to count them all, the point that **Bush**'s chief trial ***lawyer*** here, said he was trying to make in requesting that the ballots be transported here.
>
> […]

Two mentions that do not corefer share the same nominal head ("lawyer"). The coreference link can be pruned by noting that both occurrences of "lawyer" participate in an Person-Social (Business) relation, while the person name arguments of these two relation instances do not corefer ("Vice President Al Gore" vs. "Bush").

If two mentions belong to different inventors, manufacturers, user-or-owners, business groups, ethnic groups or families, or located in different places, or they are residents of different countries, employees of different organizations, or they have different ideologies, then they are less likely to corefer.

Example 5-4

> […] But the unknown culprits, who had access to some of the company's computers for an undetermined period, were not able to view or steal the company's crucial source code for its Windows or Office software, **a company *spokesman*** said Friday afternoon.
>
> Speaking earlier to Microsoft programmers and reporters at a seminar in Stockholm, Sweden, Steven Ballmer, **the company's *chief executive***, said, ``It is clear that hackers did see some of our source code," Reuters and The Associated Press reported. […]

"chief executive" stands in relation EMP-ORG/Employ-Executive with "the company", while a "company spokesman" is in relation EMP-ORG/Employ-Staff with the same company. So these two mentions are very unlikely to corefer.

Example 5-5

> […] The Texas governor and Republican nominee was described as in good spirits, but disappointed after Friday's ***Florida* Supreme Court** order for an immediate recount of so-called undervotes missed in machine tallies.
>
> …
>
> But Bush officials did little to hide their obvious dismay with Friday's 4-3 ***state* Supreme Court** ruling favoring Democrat Al Gore. […]

If we have detected a coreference link between the two "Supreme Court" mentions, as well as EMP-ORG/Subsidiary relations between this organization and two GPE mentions "Florida" and "state", it is likely that the two mentions both refer to the same state. Without this inference, it might be difficult to detect this coreference link.

Another obvious constraint which we can obtain from relation knowledge is two related mentions don't corefer. Most of such errors can be avoided by substring or head matching, but deeper semantic relation constraint can certainly help further filter the incorrect coreference links. For example, "Assad " and "Hafez Assad" don't corefer in the following sentence because they have ACE relation "Personal-Social (Family)":

Example 5-6

> […] ***Assad***, 35, has focused on domestic issues since **his father**, ***Hafez Assad***, died in June after 30 years in power.

Further progress will likely be aided by flexible frameworks for representing and using the information provided by this kind of semantic relation between mentions. This thesis uses an ontology that describes ACE relations between entities along with a training corpus annotated for relations under this ontology, the output of a relation tagger can be used to refine the results of coreference resolution.

5.1.2.2 Interaction between Coreference Resolution and Event Detection

This thesis also attempted to use event information as additional constraint rules in coreference resolution. As an example, consider a fragment from ACE data:

Example 5-7

> […] Last Saturday night, **rebels** in the Central Africa Republic, captured ***the airport in the capital city of Bangui*** and the residence of President Ange-Felix Patasse, who is out of the country.
>
> The **rebels** have captured ***the country's main international airport***, said a top official in a 300-strong African security force policing the city. […]

If we can detect "the airport in the capital city of Bangui" and "the country's main international airport" are the places involved in the "Transaction/Transfer-Ownership" event (triggered by "captured") with the same person entity "rebels", then these two airports are likely to corefer.

Example 5-8

> […] ***The Hong Kong Jockey Club*** is in talks about buying out ***the horse racing club*** in neighboring Macau, a newspaper reported Tuesday. […]

In this example, “Hong Kong Jockey Club ” is the “buyer” whereas “the horse racing club” is the “artifact” of the event “Transaction/Transfer-Ownership” (triggered by “buy out”), so these two clubs are unlikely to corefer.

Example 5-9

> […] Army forces rolled past dozens of dead Iraqi soldiers and bombed-out hulks of Iraqi military equipment as they made their way toward ***Baghdad*** from ***the area around Karbala***. […]

If we detect that “Baghdad” is the “destination” of “movement” event (triggered by “way toward”) and “the area around Karbala” is the “origin”, then this can be used as an additional feature to determine these two mentions are not coreferential.

So Event detection provides the semantic relationships between the arguments and triggers, thus allowing us to include such document-level event information into the relations holding between two candidate mentions.

5.2 Cross-lingual Interaction between Entity Extraction and Entity Translation

The interaction between two NLP tasks may allow them to share training data and resources indirectly, and thus produce better results for each. This thesis will focus on exploring such cross-task interactions between IE and machine translation.

Currently IE performance varies from language to language; for many stages English IE achieves better performance than other languages because of richer linguistic resources. Therefore, if an IE stage faces a performance bottleneck for language *S* using all the monolingual resources in *S*, it may be helpful to bridge *S* with English using machine translation, and incorporate the results of the same stage in English as useful 'feedback'.

In other words, one further source of information for improving IE are bitexts – corpora pairing the text to be tagged with its translation into one or more other languages. Such bitexts are becoming available for many language pairs, and now play a central role in the creation of machine translation and name translation systems. By aligning the texts at the word level, it's possible to infer properties of a sequence *s* in language *S* from the properties of the sequence of tokens *t* with which it is aligned in language *T*. For example, focusing on the task of improving name tagging, the basic intuition for this approach is the following: knowing that t is a name, or merely that it is capitalized (for *T* = English) makes it more likely that *s* is a name. So if there are multiple, closely competing name hypotheses in the source language *S*, the bitext can be used to select the correct analysis.

Some examples are presented as follows, for using word-aligned bitexts to improve source language (*S*) IE, with Chinese-English pair and Chinese-Japanese pair.

- ***Chinese → English***

Chinese does not have white space for tokenization or capitalization, features which, for English, can help identify name boundaries and distinguish names from nominals. Therefore using Chinese-English bitexts can help capture such indicative information to improve Chinese name tagging. For example,

Example 5-10

(a) Results from Chinese name tagger
*美德联盟立刻委任了一名执行人员出任 <ENAMEX TYPE="ORG">**三菱新**
</ENAMEX>总裁。*

(b) Bitext
Chinese: ***三菱*** ***新***
| |
English: ***Mitsubishi*** ***new***

(c) Name tagging after using bitext
*美德联盟立刻委任了一名执行人员出任 <ENAMEX TYPE="ORG">三菱
</ENAMEX>**新**总裁。*

Based on the title context word “总裁 (president)”, the Chinese name tagger mistakenly identified “Mitsubish new” as an organization name. But the un-capitalized English translation of “new” can provide a useful clue to fix this boundary error.

- ***English → Chinese***

On the other hand, Chinese has some useful language-specific properties for entity extraction. In English without global contexts sometimes it’s hard to classify an abbreviation name, but if we can translate (expand) them into Chinese full names based on bitexts, additional useful clues from the full names can be used for disambiguation.

In addition, an English name tagger heavily relying on capitalization features tends to mistakenly identify agreements, regulations, exhibitions and meetings as organization names, because they appear in the same form of capitalized strings. But in Chinese, the ‘usable-character’ feature mentioned in section 5.1.1.1 - particular character or word vocabulary for names – can be exploited as useful ‘feedback’ for fixing these errors. For example,

Example 5-11

(a) Results from English name tagger
*The flashpoint in a week of bitter <ENAMEX TYPE="ORG">**West Bank**</ENAMEX> clashes.*

(b) Bitext
English: ***West Bank***
Chinese: *西岸*

(c) Name tagging after using translation
*The flashpoint in a week of bitter <ENAMEX TYPE="LOC"> **West Bank**</ENAMEX> clashes.*

"Bank" in English can be the suffix word of either a ORG or LOC name, while its Chinese translation "岸 (shore, side)" indicates that "West Bank" is more likely to be a LOC name.

- ***Chinese → Japanese***

It's difficult to identify Japanese names in Chinese texts because of their flexible name lengths. However, if they can be 'back-translated' into Japanese, the Japanese-specific information could be used for names – they cannot include kana – to fix the name boundary:

Example 5-12

(a) Results from Chinese name tagger
*满洲时期的官员包括实业部次长<ENAMEX TYPE="PERSON">**岸信介等**</ENAMEX>。*

(b) Bitext
Chinese: ***岸信介 等***
Japanese: ***岸信介*** など

(c) Name tagging after using bitext
*满洲时期的官员包括实业部次长<ENAMEX TYPE="PERSON">**岸信介**</ENAMEX>* **等。**

The kana “など” in translation helps to fix the name boundary from “岸信介等” to”岸信介”.

The translation knowledge source has an additional benefit: because name variants in *S* may translate into the same form in *T*, translation can also aid in identifying name coreference in *S*.

5.3 Conclusion

The chapter has shown that the interaction between NLP stages can provide more comprehensive treatment of linguistic phenomena. To summarize, this thesis explores the diverse ‘feedback’ information in the following Figure 5-4.

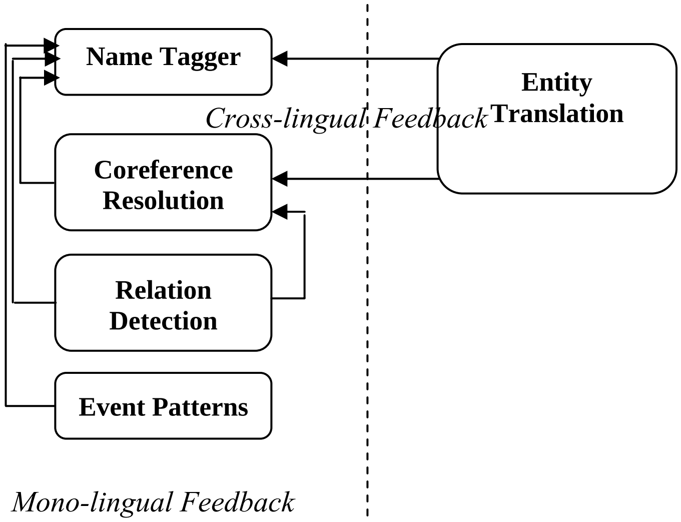


Figure 5-4. Outline of Feedback

6. INTERACTION MODELS AND ALGORITHMS

This chapter will present a new framework to efficiently and effectively incorporate the interactions in Chapter 5.

6.1 Introduction

Before introducing this new framework, this section will discuss two other possible alternative solutions in order to show the necessity of a new design.

Many current NLP stages fall into the general category of history-based noisy channel models (Brown et al., 1990; Black et al., 1993), where each stage is represented as a derivation and the probability of the possible hypothesis is then calculated. Therefore, the first natural solution is directly adding interaction knowledge as new features in these stages. However, adding global features based on the feedback from subsequent stages may require major changes of the model to take the new features into account, and also multiple runs of the entire pipeline.

Another alternative solution is to apply sequence models such as Conditional Random Fields (CRFs, Lafertty et al., 2001) and Maximum-margin Markov networks to optimize a global objective function over the space of all sequences, leveraging global features of the input. However, the main challenge in applying these joint methods more widely throughout NLP is that they are more complex and more expensive. For example, the integer linear programming framework proposed by Roth and Yih (2002, 2004, 2007) considers exhaustive hypotheses for name classification assuming the identification (the exact boundaries of entities) given, which can lead to a lot of possibilities for the real name tagging task consisting of identification and classification. (Florian et al., 2006) also pointed out that the joint model ran twice as slow as the sequential and monolithic models.

This thesis incorporates interaction knowledge using a relatively simpler mechanism. The sequential IE framework is retained, with each stage producing multiple hypotheses. Then the interactions with subsequent stages are encoded as feedback properties. Based on these properties, we apply either correction rules or re-ranking models to select the best hypothesis. This relatively simpler mechanism can allow us to apply interaction features, algorithms and scoring metric to address each stage. In addition, this staged design has the advantage of reducing the number of interaction features in any single model, thus preventing the optimization process from being overwhelmed by too many features for the amount of training data.

6.2 General Framework

6.2.1 Overview

The new NLP framework based on stage interaction is shown in Figure 6-1, and the general algorithm of incorporating interaction knowledge is presented in Figure 6-2. The central idea is to conduct inference only after the candidate results for the target stage have been narrowed down to a small set of hypotheses *H*, assuring a high probability of the best hypothesis belonging to *H* (a high performance upper-bound). This is a more pragmatic approach to filter the hypotheses in the search space to a manageable level so as to alleviate the scalability problem.

The focus of this thesis is on improving IE performance, so the different stages inside IE task are explored, and the feedback from one subsequent task – machine translation - is used as additional feedback. The next sections will describe the key modules in this framework.

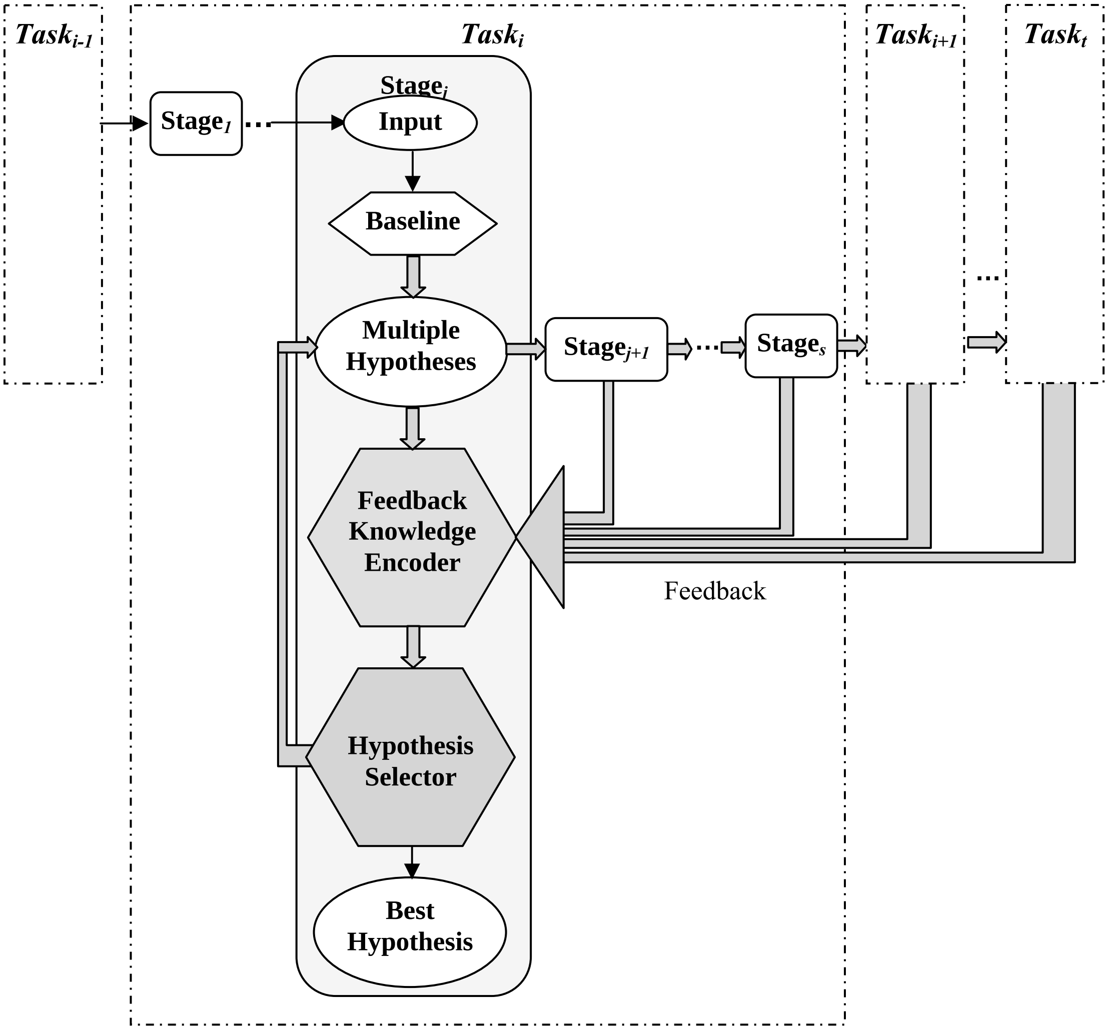


Figure 6-1. Stage-Interaction based NLP Framework

```
1. Arrange the various NLP tasks $Task_1, Task_2, ... Task_t$ in a sequential pipeline.
2. For i = 1 to t
   1) Arrange the stages $Stage_1, Stage_2, ... Stage_s$ of $Task_i$ in a sequential pipeline.
   2) For j = 1 to s
      A. Select the set of target elements (mention, entity, sentence, document, etc.)
         to improve: $Element_1, Element_2, ...Element_e$.
      B. For k = 1 to e
         a) Apply the baseline model of $Stage_j$ to generate the set of
            multiple candidate hypotheses $H = \{h_1, h_2, ..., h_n\}$ for $Element_k$
         b) Process each hypothesis in $H$ through subsequent stages
            $Stage_{j+1}, ...Stage_s$, or subsequent tasks $Task_{i+1}, ...Task_t$,
            and get results $H\_SubsequentResult$
         c) Select feedback knowledge $H\_Feedback$ from $H\_SubsequentResult$
         d) Use a rule-based or supervised learning based module incorporating
            $H\_Feedback$ to determine the best hypothesis $h_{best}$ from $H$; In the rule-
            based approach correct $H\_SubsequentResult$ and then go back to (c) if
            necessary.
         e) Output $h_{best}$ to the next stage or task.
```

Figure 6-2. Stage Interaction based Correction Algorithm

6.2.2 Target Element and Multiple Hypotheses Representation

The general goal of this framework is to accurately select the assignment that maximizes preference of a target element. In our application, the choice of such 'element' can be customized based on the stage we aim to improve., such as entity detection, relation detection, or name tagging. Table 6-1 presents some examples of target element and their hypotheses for different stages.

6.2.3 Feedback Knowledge Selection

Despite the intuition that linguistically-derived sophisticated interaction knowledge should be beneficial to IE, there have been few reliable demonstrations of real gains in performance. It's not sufficient to directly augment the feature space of a baseline stage with all the feedback results from subsequent stages and tasks. Often, such a direct approach raised performance issues because these interaction representations are not amenable to large-scale feature engineering. More importantly, subsequent stages in the pipeline can introduce much noise unless features are carefully selected.

Therefore, in order to apply the interaction knowledge more effectively, the careful selection and engineering of features will be needed. It will be necessary to recognize the situations where we can expect to improve performance, and encode them in a robust and controlled manner that is known to deliver useful additional information.

The new framework presented in Figure 6-2 selectively uses and quantifies the interaction information which has high reliability, and provide confidence measures for additional global features which otherwise would be unreliable to be used in the baseline stages. For example, when using the feedback from coreference to improve name tagging, instead of using directly the results of whether two mentions are coreferred or not as features, a "*CorefNum*" feature is encoded: the names in one hypothesis are referred to by *CorefNum* other mentions, because this feature is more likely to reliably help in selecting the best hypothesis. More details about the selection of feedback knowledge will be presented in the case studies in Chapter 7 and 8.

Target Stage	Target Element	Multiple Hypotheses (Best Hypothesis *)
Name Tagging	Sentence *"Slobodan Milosevic was born in Serbia."*	h_0: *Slobodan Milosevic was born in <PER>Serbia</PER> .* h_1: *<PER>Slobodan</PER> Milosevic was born in <GPE>Serbia</GPE> .* *h_2: *<PER>Slobodan Milosevic</PER> was born in <GPE>Serbia</GPE>.* …
Coreference Resolution	Mention-Antecedent Pair *"the airport in the capital city of Bangui" –"the country's main international airport"*	*h_0: *"the airport in the capital city of Bangui" and "the country's main international airport" corefer.* h_1: *"the airport in the capital city of Bangui" and "the country's main international airport" don't corefer.*
Entity Extraction	Cross-lingual Entity Pair $\begin{pmatrix}\text{Source Mention}_1\\ \text{Source Mention}_2\\ \cdots\end{pmatrix}-\begin{pmatrix}\text{Target Mention}_1\\ \text{Target Mention}_2\\ \cdots\end{pmatrix}$	h_0: $\begin{pmatrix}\text{阿贾比}\\ \text{阿贾比由}\\ \text{哈米德.阿贾比}\\ \text{阿贾比}\end{pmatrix}-\begin{pmatrix}\textit{Agabi}\\ \textit{Agabi from}\\ \textit{Hamid Agabi}\\ \textit{Agabi}\end{pmatrix}$ *h_1: $\begin{pmatrix}\text{阿贾比}\\ \text{阿贾比}\\ \text{哈米德.阿贾比}\\ \text{阿贾比}\end{pmatrix}-\begin{pmatrix}\textit{Agabi}\\ \textit{Agabi}\\ \textit{Hamid Agabi}\\ \textit{Agabi}\end{pmatrix}$ h_2: $\begin{pmatrix}\text{阿贾比由}\\ \text{阿贾比由}\\ \text{哈米德.阿贾比由}\\ \text{阿贾比由}\end{pmatrix}-\begin{pmatrix}\textit{Agabi from}\\ \textit{Agabi from}\\ \textit{Hamid Agabi from}\\ \textit{Agab from}\end{pmatrix}$

Table 6-1. Examples for Target Element and Multiple Hypotheses

6.3 Inference Details

Now the critical question becomes: how to use a suitable method to incorporate the interaction information?

Two different approaches are examined and compared: rule-based and re-ranking. The following sections shall describe their motivations and implementations in detail and compare their requirements and characteristics, in order to understand which methods are best to provide the greatest improvement in IE on particular stages.

6.3.1 Rule-based System

The simplest way to exploit the interaction knowledge sources is to convert them into correction rules, and apply them in the preprocessing or post-processing phase of the baseline model. A reliability weight can be assigned to each rule to determine the rule's accuracy, and then apply the constructed rules to adjust (filter/recover) or correct the sequential baseline outputs.

For example, a correction rule can be encoded for using relation detection results to filter out coreference links, "If person Mention_1 is an executive member of a company, and person Mention_2 is a staff member of the same company, then Mention_1 and Mention_2 do not corefer". The translation feedback can be encoded to fix source language entity detection by heuristic rules such as "If Mention_1 is translated into lower case in English with high confidence, then change its mention type to nominal".

6.3.2 Shift from Rules to Re-Ranking

Correction rules don't require any labeled data and can be applied to all granularities of hypotheses; for example, for the problem of name tagging, a hypothesis can be a candidate name mention, an entity consisting of coreferred names, a sentence with an alternative name labeling, or even a document labeled with name mentions. However, the rule-based approach has the following limitations:

- Rules need to be encoded with very high accuracy.
- Rules have to be formulated in condition-action form.
- Much engineering effort and language specific knowledge are needed for adjusting the confidence values, thresholds and the priority order of different rules

- The binary action based on rules is not suitable for all circumstances. Some estimate of the correctness of a hypothesis will be needed, to give some rough indication of the confidence in the extraction of a particular piece of information.

Many current IE stages are based on supervised learning models trained from hand-labeled corpora. Such valuable labeled data can be re-used to learn automatically the coefficients to combine the feedback knowledge. However, it’s not feasible to simply assign constant weights and linearly combine them because for the following two reasons:

- Weights can vary depending on training and test data.
- The feature space may not be linearly separable; some interaction features may overlap.

This thesis employs supervised re-ranking models as an alternative method to incorporate the interactions. Each stage generates N-Best hypotheses, and then the feedback information is used to re-rank these hypotheses. Besides addressing the limitations of the rule-based approach mentioned above, the re-ranking approach is suitable for the system described in the thesis for the following reasons.

Most stages in the English and Chinese baseline IE systems described in Chapter 4 are built on statistical models, so they are well suited to producing and maintaining multiple hypotheses about a sentence. For example, the HMM name tagger can produce multiple hypotheses in the form of N-Best lists; the MaxEnt-based coreference resolver can produce all possible positive coreference links.

Another advantage of statistical baselines is that additional features comparing the hypotheses can be propagated through the pipeline, which includes the probability of each hypothesis, the margin between the first and second hypotheses, the voting rate among all hypotheses, etc. All of them can be exploited as baseline confidence features in the re-ranking model.

6.3.3 Re-Ranking

This section shall first describe the general setting and the special characteristics of re-ranking, and show how an approach based on multi-stage incremental re-ranking can effectively handle features across sentence and document boundaries. Then three re-ranking models are presented– MaxEnt-Rank, SVMRank and p-Norm Push Ranking,

6.3.3.1 Overview

The general procedure of re-ranking is as follows.

- ❑ Use baseline to generate N-Best hypotheses and initial ranking/confidence.
- ❑ Build a second-phase supervised ranking model.
- ❑ Encode the baseline output confidence and feedback knowledge as features.
- ❑ Predict new rankings for the hypotheses, generate the new top hypothesis as the final output.

For example, in the name re-ranking model, each hypothesis is name annotation of the entire sentence; for the sentence "*Slobodan Milosevic was born in Serbia*", the following hypotheses may be generated from the baseline model:

- h_0: *Slobodan Milosevic was born in <PER>Serbia</PER> .*
- h_1: *<PER>Slobodan</PER> Milosevic was born in <GPE>Serbia</GPE> .*
- h_2: *<PER>Slobodan Milosevic</PER> was born in <GPE>Serbia</GPE> .*

…

Then the goal of re-ranking is to assign new ranking scores for these hypotheses and push the new best hypothesis to the top of the list:

- h_0: *<PER>Slobodan Milosevic</PER> was born in <GPE>Serbia</GPE>* .
- h_1: *<PER>Slobodan</PER> Milosevic was born in <GPE>Serbia</GPE>* .
- h_2: *Slobodan Milosevic was born in <PER>Serbia</PER>* .

6.3.3.2 Training Hypotheses Generation

An important step for supervised learning models is to obtain training data. The training data for re-ranking can be constructed directly from the labeled corpora for the baseline stage.

Assume a training corpus (*D*) is labeled with reference IE annotations;, this thesis adopts two alternative ways – partial training and k-fold training - to obtain the training data (*RD*) for the re-ranking algorithm. The details are presented in Figure 6-3.

In K-fold training, all the labeled baseline corpora can be utilized to obtain more training data for re-ranking, although the training procedure becomes much slower then partial training.

6.3.3.3 Training and Test Algorithm

The general re-ranking algorithm is shown in Figure 6-4. More details about sample creation and re-ranking models will be presented in the next subsections.

```
Partial Training
1. Separate the training data $D$ into two portions $D_1$ and $D_2$.
2. Train a baseline model $BM$ on $D_1$.
3. For each sentence in $D_2$
    (1) Apply $BM$ to generate multiple hypotheses $H = \{h_1, h_2,...,h_n\}$.
    (2) For each $h_i$ in $H$
        Measure the performance of $h_i$ against the key in the annotated corpus and get its score $s_i$. Add the pair $< h_i , s_i>$ into training data $RD$.

K-Fold Training
1. Split the training corpus $D$ into k folds $\{D_1, D_2 ... D_k\}$.
2. For i= 1 to k
    (1) Train a baseline model $BM_i$ on $\{D_1, \ldots D_{i-1}\}$ and $\{D_{i+1}, \ldots D_k\}$.
    (2) For each sentence in $D_i$
        (a) Apply $BM_i$ to generate multiple hypotheses $H = \{h_1, h_2,...,h_n\}$.
        (b) For each $h_i$ in $H$
            Measure the performance of $h_i$ against the key in the annotated corpus and get its score $s_i$. Add the pair $< h_i, s_i>$ into training data $RD$.
```

Figure 6-3. Re-Ranking Training Hypotheses Generation

```
Training
1. Train a baseline stage $BM$ which can generate N-Best hypotheses for each sentence and produce a probability associated with each hypothesis.
2. Apply $BM$ to generate N-Best hypotheses $H_{train}$ for each sentence.
3. Process each hypothesis $h_{(i,train)}$ in $H_{train}$ through subsequent stages or tasks.
4. Create training samples $S_{train}$ from $H_{train}$
5. For each training sample $s_{(i,,train)}$ in $S_{train}$, select inference features $F_{train}$ from the output of step 3.
6. Train a statistical re-ranking model $RRM$ based on each $s_{(i,,train)}$, using $F_{train}$ together with the probability from the baseline stage.

Test
1. Apply $BM$ to generate N-Best hypotheses $H_{test}$ for each test sentence.
2. Process each hypothesis $h_{(i, test)}$ in $H_{test}$ through subsequent stages or tasks.
3. Create test samples $S_{test}$ from $H_{test}$
4. For each test sample $s_{(i, test)}$ in $S_{test}$, from the output of step 2 encode feature set $F_{test}$ with the same types as $F_{train}$.
5. Apply $RRM$ on $S_{test}$ to determine a new ranking for $H_{test}$
6. Output $h_{best}$ on the top of the re-ranked $H_{test}$.
```

Figure 6-4. Re-Ranking Training and Testing

6.3.3.4 Sample Creation

Two different sampling methods – Single Sampling and Pair-wise Sampling - are adopted to create re-ranking training and test samples.

- **Single Sampling**

The first approach is to use each single hypothesis h_i as a sample. Only the best hypothesis of each sentence is regarded as a positive sample; all the rest are regarded as negative samples. In general, absolute values of features are not good indicators of whether a hypothesis will be the best hypothesis for a sentence; for example, a co-referring mention count of 7 may be excellent for one sentence and poor for another. Consequently, in this single-hypothesis-sampling approach, each feature is converted to a boolean value, which is true if the original feature takes on its maximum value (among all hypotheses) for this hypothesis. This does, however, lose some of the detail about the differences between hypotheses.

- **Pair-wise Sampling and Pruning**

In pair-wise sampling each pair of hypotheses (h_i, h_j) is used as a sample. The value of a feature for a sample is the difference between its values for the two hypotheses.

However, considering all pairs causes the number of samples to grow quadratically ($O(N^2)$) with the number of hypotheses, compared to the linear growth with best/non-best sampling. To make the training and test procedures more efficient, the data is pruned in several ways. Pruning is performed by beam setting, removing candidate hypotheses that possess very low probabilities from the baseline, and during training the hypotheses are discarded with very low performance. The pairs very close in performance or probability are also discarded.

Additionally, the Relative Pruning technique (Chiang 2005) is used, by which any hypothesis h_i is discarded if:

Prob(h_i) < α (the highest probability for the hypothesis set H).

A "crucial pair" is defined as a pair of hypotheses such that, according to their performance, the first hypothesis in the pair should be more highly ranked than the second. That is, if for a sentence, the performance of hypothesis h_i is larger than that of h_j, then (h_i, h_j) is a crucial pair.

6.3.3.5 Learning Functions

This thesis investigated the following four learning functions for the re-ranking problem.

- **Score Based Direct Re-Ranking (SDRR)**

For each hypothesis h_i, learn a scoring function $f: H \rightarrow R$, such that $f(h_i) > f(h_j)$ if the performance of h_i is higher than the performance of h_j. Then select the hypothesis which achieves the largest value for $f(h_i)$.

- **Classification Based Direct Re-Ranking (CDRR)**

For each hypothesis h_i, learn f: $H \rightarrow \{-1, 1\}$, such that $f(h_i) = 1$ if h_i has the top performance among H; otherwise $f(h_i)$ = -1. Then select the hypothesis which achieves the largest value for $prob\ (f(h_i))$.

- **Pairwise-Comparison Indirect Re-Ranking (PIRR)**

For each "crucial" pair of hypotheses (h_i, h_j), learn $f : H \times H \rightarrow \{-1, 1\}$, such that $f(h_i, h_j) = 1$ if h_i is better than h_j; $f\ (h_i, h_j) = -1$ if h_i is worse than h_j. This is called as "indirect" ranking because an additional decoding step is needed to pick the best hypothesis from these pair-wise comparison results. An example of the decoding algorithm will be described in section 6.4.5.5.1.

- **Baseline-Comparison Indirect Re-Ranking (BIRR)**

For each hypothesis h_i *(i>0)* learn $f : H \rightarrow \{-1, 1\}$, such that $f(h_i) = 1$ if h_i is better than h_0; $f(h_i)$ = -1 if h_i is worse than h_0, where h_0 is the top hypothesis produced by the baseline system. If for all *i (i>0)* $prob(f(h_i) = 1) <= 0.5$, then select h_0 as the best hypothesis; otherwise select the hypothesis which achieves highest $prob(f(h_i) = 1)$. The advantage of this model is that it can still benefit from pair-wise feature encoding, while avoiding the decoding step of resolving ambiguities as in PIRR.

In the CDRR, PIRR and BIRR frameworks, the classification values can be further scaled into multiple variants. For example, one can have f: $H \rightarrow$ {-2, -1, 1, 2}; in CDRR it can indicate how close the performance of h_i is to the top performance among H; in BIRR it can help distinguish whether h_i is much better than h_0 or they perform equally well.

Table 6-2 presents a simple example of these ranking functions for $H= \{h_0, h_1, h_2\}$.

Hypothesis	Score (%)	SDRR	CDRR	PIRR	BIRR
h_0 h_1 h_2	80 90 60	$f(h_0) = 0.8$ $f(h_1) = 0.9$ $f(h_2) = 0.6$	$f(h_0) = -1$ $f(h_1) = 1$ $f(h_2) = -1$	$f(h_0, h_1) = -1$ $f(h_0, h_2) = 1$ $f(h_1, h_0) = 1$ $f(h_1, h_2) = 1$ $f(h_2, h_0) = -1$ $f(h_2, h_1) = -1$	$f(h_1) = 1$ $f(h_2) = -1$

Table 6-2. Example for Re-Ranking Functions

6.3.3.6 Multi-Stage Incremental Re-Ranking

If we revisit the framework in Figure 6-1 regarding re-ranking implementation, we can see that the cross-sentence and cross-document interaction constraints will give rise to new design issues.

For example, coreference is potentially a powerful contributor for enhancing NE recognition, because it provides information from other sentences and even documents, and it applies to all sentences that include names. For a name candidate, 62% of its coreference relations span sentence boundaries. However, this breadth poses a problem because it means that the score of a hypothesis for a given sentence may depend on the tags assigned to the same names in other sentences.[5]

Ideally, when re-ranking the hypotheses for one sentence *S*, the other sentences that include mentions of the same name should already have been re-ranked, but this is not possible because of the mutual dependence. Repeated re-ranking of a sentence would be time-consuming, so the following alternative approach has been adopted.

Note that it’s not needed to do as elaborate a re-ranking after each stage, since the ranking result at each stage doesn’t have to be precise; as long as each stage can keep the correct one in the top N hypotheses, at a high confidence level. Therefore, instead of incorporating all interaction information in one re-ranker, two re-rankers are applied in succession.

In the first re-ranking step, new rankings are generated for all sentences based on all interaction knowledge which can be obtained within sentences. Then in a second pass, a

[5] For in-document coreference, this problem could be avoided if the extraction of an entire document constituted a hypothesis, but that would be impractical … a very large N would be required to capture sufficient alternative extractions in an N-best framework.

re-ranker is applied based on cross-sentence interaction between the candidate hypothesis of sentence *S* and the top-ranking hypothesis (from the first re-ranker) of all other sentences.[6] In this way, the second re-ranker can propagate globally (across sentences and documents) high-confidence decisions based on the other evidence.

At each re-ranking step the algorithm will generate the best name hypothesis directly and by-pass later re-ranking steps if the re-ranker has high confidence in its decisions. Otherwise the sentence is forwarded to the next re-ranker, based on other features. In this way the algorithm can adjust the ranking of multiple hypotheses and seek the best tagging for each sentence gradually.

6.3.3.7 Learning Models

It's worth trying several learning models to see if one does better to fit our re-ranking application.. There are many learning schema (MaxEnt, SVM, Boosting, etc.) and particular data properties may make some models work better than others. Some models have provable properties, for example regarding generalization error. This thesis chooses three state-of-the-art ranking algorithms that have good generalization ability: MaxEnt-Rank, SVMRank, and p-Norm Push Ranking. The following sections will describe these algorithms. In Chapter 7 these algorithms will be applied and evaluated for re-ranking in the context of name tagging.

- **MaxEnt-Rank**

Maximum Entropy (MaxEnt) models are useful for the task of ranking because they compute a reliable ranking probability for each hypothesis. During training the Pairwise-

[6] This second pass is skipped for sentences for which the confidence in the top hypothesis produced by the first re-ranker is above a threshold.

Comparison Indirect Re-Ranking (PIRR) model is used to learn the ranking function *f*. During test the MaxEnt model produces a probability for each un-pruned "crucial" pair: $prob(f(h_i, h_j) = 1)$, i.e., the probability that for the given sentence, h_i is a better hypothesis than h_j.

An additional decoding step is needed to select the best hypothesis. Inspired by the caching idea and the multi-class solution proposed by (Platt et al. 2000), this thesis uses a dynamic decoding algorithm with complexity O(n), as shown in Figure 6-5.

```
Prune
 for i = 1 to n
    Num = 0;
    for j = 1 to n and j≠i
         if CompareResult(h_i, h_j) = "worse"
              Num++;
  if Num> β then discard h_i from H
Decoding
 Initialize: i = 1, j = n
 while (i<j)
     if CompareResult(h_i, h_j) = "better"
        discard h_j from H;
        j--;
     else if CompareResult(h_i, h_j) = "worse"
        discard h_i from H;
        i++;
     else break;
Output
If the number of remaining hypotheses in H is 1, then output it as the best
hypothesis; else propagate all hypothesis pairs into the next re-ranker.
```

Figure 6-5. MaxEnt-Rank Decoding

The probability values are scaled into three types: *CompareResult*(h_i, h_j) = “better” if $prob(f(h_i, h_j) = 1) > \delta_1$, “worse” if $prob(f(h_i, h_j) = 1) < \delta_2$, and “unsure” otherwise, where $\delta_1 \geqslant \delta_2$.[7]

- **SVMRank**

A Support Vector Machines (SVMs, Cristianini and Shawe-Taylor, 2000) based model is also applied, which can theoretically achieve very low generalization error by emphasizing correct interactions while ignoring noisy ones. SVMRank uses the Pair-wise Sampling scheme and PIRR ranking function as for MaxEnt-Rank.

In addition the following adaptations are made: the SVM outputs are calibrated, and the data is separated into subsets. To speed up training, our training samples are divided into *k* subsets. Each subset contains *N(N-1)/k* pairs of hypotheses of each sentence.

The output of an SVM yields a distance to the separating hyperplane, but not a probability. This thesis has applied the method described in (Shen and Joshi, 2003), to map SVM’s results to probabilities via a sigmoid. Thus from the k^{th} SVM, the probability for each pair of hypotheses is generated:

$$prob(f_k(h_i, h_j) = 1),$$

namely the probability of h_i being better than h_j. Then combining all *k* SVMs’ results:

$$Z(h_i, h_j) = \prod_k prob(f_k(h_i, h_j) = 1).$$

So the hypothesis h_i with maximal value is chosen as the top hypothesis:

$$\arg\max_{h_i}(\prod_j Z(h_i, h_j)).$$

[7] In the final stage re-ranker we use $\delta_1 = \delta_2$ so that we don’t generate the output of “unsure”, and one hypothesis is finally selected.

- **p-Norm Push Ranking**

The third algorithm is a general boosting-style supervised ranking algorithm called p-Norm Push Ranking (Rudin, 2006). This algorithm is a generalization of RankBoost (Freund et al. 1998, take p=1 for RankBoost) which concentrates specifically on the top portion of a ranked list. This algorithm is very efficient and can be turned into a margin-maximization algorithm in the same way as soft margin RankBoost.

The parameter "p" determines how much emphasis (or "push") is placed closer to the top of the ranked list, where $p \geq 1$. When p is set at a large value, the rankings at the top of the list are given higher priority (a large "push"), at the expense of possibly making mis-ranks towards the bottom of the list. The applications in this thesis do not care about the rankings at the bottom of the list (i.e., do not care about the exact rank ordering of the bad hypotheses), so this algorithm is suitable.

There is a tradeoff for the choice of p; larger p yields more accurate results at the very top of the list for the training data. If the application considers more than simply the very top of the list, a smaller value of p may be desired. Note that larger values of p also require more training data in order to maintain generalization ability (as shown both by theoretical generalization bounds and experiments).

If a large p is desired, the value of p must still be limited in order to allow generalization, given the amount of training data.

The objective of the p-Norm Push Ranking algorithm is to create a scoring function f in the way of SDRR ranking function as described in section 6.3.3.5, namely:

$f: H \rightarrow R$ such that for each crucial pair $(h_i, h_j), f(h_i) > f(h_j)$.

The form of the scoring function is:

$f(h_i) = \sum \alpha_k g_k(h_i)$,

where g_k is called a weak ranker: $g_k : H \rightarrow [0,1]$. The values of α_k are determined by the p-Norm Push algorithm iteratively.

The weak rankers g_k use the feedback features from subsequent stages. Note that the algorithm is allowed to use both g_k and $g'_k(h_i)=1-g_k(h_i)$ as weak rankers, namely when g_k has low accuracy on the training set; this way the algorithm itself can decide which to use.

As in the style of boosting algorithms, real-valued weights are placed on each of the training crucial pairs, and these weights are successively updated by the algorithm. Higher weights are given to those crucial pairs that were misranked at the previous iteration, especially taking into account the pairs near the top of the list. A price is put on each negative sample h_i:

$$G\left(\sum_{i=1}^{I} 1_{f(h_i) \leq f(h_j)}\right),$$

where $G : R_+ \rightarrow R_+$ is a convex, monotonically increasing function, such as the function used in this thesis: $G(z) = z^p$ for p large. At each iteration, one weak ranker g_k is chosen by the algorithm, based on the weights. The coefficient α_k is then updated accordingly.

6.3.4 Conclusion

One appeal of the re-ranking methods is their flexibility in exploiting feedback features from subsequent stages into a model: essentially any features which might be useful in discriminating good from bad hypotheses can be included. By capturing the quantified comparison results among hypotheses, these methods can be more effective when the best hypothesis exists in a relatively large set of alternative hypotheses. They can capture and incorporate global features in a natural and efficient manner. To summarize, Table 6-3 lists the main potential conditions for applying these three different joint inference approaches.

Hypothesis Selection Characteristics	Inference Rules	Re-Ranking
Require training data	No	Yes
Require explicit N-Best hypotheses Representation	No	Yes
Require logical form inference knowledge	Yes	No
Can quantify inference results and confidence	No	Yes
Can easily incorporate large amount of realistic wider context interaction features	No	Yes
Can easily incorporate features for discriminating good from bad hypotheses	No	Yes

Table 6-3. Condition Comparison for Rule and Re-Ranking based Hypothesis Selection

7. CASE STUDY ON MONOLINGUAL INTERACTION

This chapter will demonstrate the idea of mono-lingual interactions by two case studies:

- Improving name tagging by incorporating feedback from subsequent IE stages: coreference resolution, relation detection and event extraction.
- Improving coreference resolution using results from relation detection.

7.1 Improving Name Tagging by Subsequent IE Stages

The re-ranking approach is used to improve English and Chinese name tagging. N-Best name hypotheses are generated, then the results from subsequent IE stages are incorporated into re-ranking models. F-measure is used to measure the quality of each name hypothesis against the key.

Section 7.1.1 will describe the generation of N-Best hypotheses, and section 7.1.2 will describe the diverse features used in re-ranking. Then section 7.1.3 and 7.1.4 will present the experimental results, analyze the results in terms of different types of name identification and classification errors, compare three re-ranking models, and show the benefit of multi-stage re-ranking for cross-sentence and cross-document inference.

7.1.1 N-Best Hypotheses Generation

The baseline HMM name tagger produces the N-Best hypotheses for each sentence. In order to decide when we need to rely on global (coreference and relation) information for name tagging, we want to have some assessment of the confidence that the name tagger has in the first hypothesis. In this thesis, the *margin* metric described in section 5.1.1.1 is used for this purpose.

Using cross-validation on the training data, the algorithm adjusts the number of hypotheses (*N*) that the baseline tagger will generate and store through the pipeline, as a function of the margin, in order to maintain efficiency while minimizing the chance of losing a high-quality hypothesis. The margin is then divided into ranges of values, and set a value of *N* ranging from 1 to 30.

7.1.2 Re-Ranking Features

This section will present the detailed feature encoding to capture the feedback properties to improve name tagging as described in Section 4. The second column of Table 7-1 marks the language(s) for which each feature was applied.

For each pair of hypothesis (h_i, h_j), a feature set is constructed for assessing the ranking of h_i and h_j. Based on the information obtained from inferences, the property score PS_{ik} is computed for each individual name candidate N_{ik} in h_i; some of these properties depend also on the corresponding name tags in h_j. Then sum over all names in each hypothesis h_i:

$$PS_i = \sum_k PS_{ik}$$

Finally the quantity $(PS_i – PS_j)$ is used as the feature value for a pair of hypotheses (h_i, h_j) to determine whether this sum is larger for h_i or h_j. The results of these comparisons are used as features in assessing the ranking of h_i and h_j. Table 7-1 summarizes the property scores PS_{ik} used in the different re-rankers for English and Chinese name tagging.

Source	Language	Property for comparing names N_{ik} and N_{jk}	
Baseline	English & Chinese	*HMMMargin*	scaled margin value from HMM
	Chinese	*VotingRate*$_{ik}$	the voting rate for N_{ik} among all the candidate hypotheses
Name Structure	Chinese	*Idiom*$_{ik}$	-1 if N_{ik} is part of an idiom; otherwise 0
	Chinese	*ORGSuffix*$_{ik}$	1 if N_{ik} is tagged as ORG and it includes a suffix word; otherwise 0
	Chinese	*PERChar*$_{ik}$	-1 if N_{ik} is tagged as PER without family name, and it does not consist entirely of transliterated person name characters; otherwise 0
	Chinese	*TitleStructure*$_{ik}$	-1 if N_{ik} = title word + family name while N_{jk} = title word + family name + given name; otherwise 0
	Chinese	*Digit*$_{ik}$	-1 if N_{ik} is PER or GPE and it includes digits or punctuation; otherwise 0
	Chinese	*AbbPER*$_{ik}$	-1 if N_{ik} = little/old + family name + given name while N_{jk} = little/old + family name; otherwise 0
	Chinese	*SegmentPER*$_{ik}$	-1 if N_{ik} is GPE (PER)* GPE , while N_{jk} is PER*; otherwise 0
Local Context	Chinese	*PERContext*$_{ik}$	the number of PER context words if N_{ik} and N_{jk} are both PER; otherwise 0
	English	*LOCGPEFAC Context*$_{ik}$	If there is preposition in the local cotext, 1 if N_{ik} is tagged as LOC/GPE/FAC and -1 if other name types; otherwise 0
Gazetteer	English & Chinese	*GazetteerName*$_{ik}$	1 if N_{ik} is tagged as the same type in one of the gazetteer name lists; otherwise 0
Relation	Chinese	*Relation Constraint*$_{ik}$	If N_{ik} is in relation R (N_{ik} = EntityType$_1$, M$_2$ = EntityType$_2$), compute Prob(EntityType$_1$|EntityType$_2$, R) from training data; otherwise 0
	English	*Conjunction of InRelation*$_i$ *& Probability1*$_i$	*Inrelation*$_{ik}$ is 1 if N_{ik} and N_{jk} have different name types, and N_{ik} is in a definite relation while N_{jk} is not; otherwise 0.
Event	Chinese	*Event Constraint*$_i$	1 if all entity types in h$_i$ match event pattern, -1 if some do not match, and 0 if the argument slots are empty
	English	*Event Cooccurrence*$_i$	The probability of the name type and a verb appears together in a nine-word window
	Chinese	*EventSubType*	Event subtype if the patterns are extracted from ACE data, otherwise "None"
Co-reference	Chinese	*Head*$_{ik}$	1 if N_{ik} includes the head word of name; otherwise 0
	English & Chinese	*CorefNum*$_{ik}$	the number of mentions which corefer to N_{ik}
	English & Chinese	*WeightNum*$_{ik}$	the sum of all link weights between N_{ik} and its corefered mentions, assign 1 for same name-name coreference, 0.8 for different name-name coreference; 0.5 for apposition; 0.3 for other name-nominal coreference
	Chinese	*HCorefNum*$_{ik}$	the number of mentions which corefer to N_{ik} and output by previous re-rankers with high confidence
	English	*ECorefNum*$_i$	the number of entities coreferred to the name candidates in h_i

Table 7-1. Name Re-Ranking Properties

The following subsections shall describe the features of the individual re-ranking stages in further detail. Section 7.1.2.1 introduces the confidence features derived from the baseline tagger, sections 7.1.2.2 to 7.1.2.4 describe the local features which were not explicitly captured by the baseline, and then the remaining sections present the feedback features.

7.1.2.1 Baseline Confidence Features

This section considers some features for gauging the confidence of name tags assigned by the baseline HMM tagger.

- **Margin**

A large margin indicates greater confidence that the first hypothesis is correct. Figure 7-1 shows the position of the best hypothesis generated by the Chinese baseline tagger according to different values of margins, given N=20.

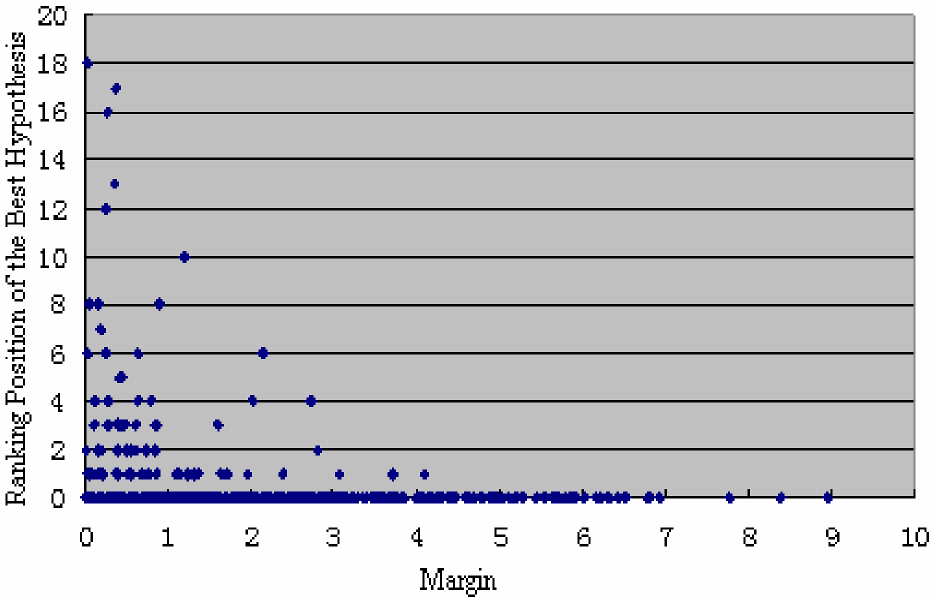


Figure 7-1. The Ranking Position of Best Hypothesis vs. Margin

We can see that as the value of margin increases, more of the first hypotheses are the best analysis results. A large margin indicates greater confidence that the first hypothesis is correct. So if the margin of a sentence is above a threshold, the first hypothesis is selected, dropping the others and by-passing the re-ranking. Scheffer et al. (2001) used a similar method to identify good candidates for tagging in an active learner.

- **Voting Rate**

In addition, the mechanism of weighted voting among hypotheses (Zhai et al., 2004) is used as an additional feature in the first-stage re-ranking. This approach allows all hypotheses to vote on a possible name output. A recognized name is considered correct only when it occurs in more than 30% of the hypotheses (weighted by their probability). The log probability produced by the HMM, $prob_i$ is used for hypothesis h_i . This probability weight is normalized as:

$$W_i = \frac{\exp(prob_i)}{\sum_q \exp(prob_q)}$$

For each name mention N_{ik} in h_i, define:

$Occur_q(N_{ik})$ = 1 if N_{ik} occurs in h_q; otherwise 0.

Then its voting value is counted as follows:

$Voting_{ik}$ = 1 if $\sum_q W_q \times Occur_q(N_{ik})$ >0.3; otherwise 0.

Finally get the voting rate of h_i: $Voting_i = \sum_k Voting_{ik}$

7.1.2.2 Name Structure Features

Ten features are included to capture Chinese name structure parsing evidence. For instance, penalizing candidate names overlapped with idiom words; giving credit to an organization name that includes an explicit, organization-indicating suffix ("Russian Nuclear Power Instituition" is more likely to be a correct name than "Russian Nuclear Power"). In some other features, higher property values are given to candidates matching possible name structures. For example, a common Chinese name tagging mistake gives rise to the sequence "GPE (PER)*GPE", because transliterated GPE and PER names share part of the character lists. But this is an impossible name structure in Chinese, so a penalty is assigned to such a sequence if it appears in the candidate hypothesis.

7.1.2.3 Local Context Features

The ranking results and confidence values from the baseline may not emphasize some important local features. So the re-ranking stage re-uses the following two indicative local features.

- **Person Context**

In Chinese name tagging one of the most difficult challenges is detecting person name boundary because the characters in the candidate name together with context can compose a common word. We re-use the person context information as a re-ranking feature, to give credit for person names with particular contexts such as person titles and verbs occurring with people's names.

- **GPE/LOC/FAC Context**

Prepositions in local context can help distinguish GPE/LOC/FAC names from other types. So the algorithm checks whether there is a preposition in the five-token-window around the name candidate, and gives credit to GPE/LOC/FAC names while penalizing other types. For example, this feature can help confirm that "*Petit Palais*" is not a person name in the sentence "*Gaspard Yurkievitch, a young French designer, showing in the Petit Palais, could make the grade in some stores with his offhand styles that some find cool.*"

7.1.2.4 Gazetteer Features

The high-frequency name lists are collected from the training corpus, country/province/state/ city lists from Chinese wikipedia, and an English organization name list (Sekine and Nobata, 2004) including 20061 entries. If a name candidate is identified as the same type in one of these lists, the confidence for the corresponding hypothesis is increased.

7.1.2.5 Relation Features

The relation and event re-ranking features are based on matching patterns of words or constituents. They serve to correct name head boundary errors (because such errors would prevent some patterns from matching). Because they exert selectional preferences on their arguments, they also correct name type errors.

The information of "in relation or not" can be used as a measure of confidence. In addition the relation patterns can be encoded as constraints for name type classification.

For each relation argument, a feature is included to represent the likelihood that relation appears with an argument of that name type. To formalize,

if N_{ik} is involved in relation $R(ARG1= N_{ik}, ARG2=M_j)$, and

if $EntityType(N_{ik}) = EntityType_1, EntityType(M_2)= EntityType_2$,

then $Prob(EntityType_1 \mid EntityType_2, R)$ is computed from relation training data.

These probabilities are scaled into an additional indicator for N_{ik} being identified as the correct entity type:

- ❑ 1: *EntityType₁* is very likely correct;
- ❑ 0.5: *EntityType₁* is likely correct;
- ❑ 0: unsure
- ❑ -0.5: *EntityType₁* is unlikely correct;
- ❑ -1: *EntityType₁* is very unlikely correct

For example, the name N_{ik} matching the condition *(PER | PER, Per_Social)* is very likely to be correct; while *(ORG | PER, Per_Social)* is very unlikely to be correct, etc. This information helps to select the hypothesis with correct name type recognition.

7.1.2.6 Event Features

For Chinese the algorithm also compares the types of the names filling argument slots with the entity types required by the event pattern, and assigns a score which is positive if all entity types match, negative if some do not match, and zero if the argument slots are empty. Only 11% of the sentences in the test data contain instances of the ACE event types. To increase the impact of the event patterns, additional frequent event trigger words are included from Chinese PropBank, so that finally 35% of sentences contain event "trigger words". A boolean flag is encoded to distinguish whether the pattern is extracted from ACE data.

For English, for each name in the ACE training data the algorithm checks whether there is a verb *v* in its nine-token-wide window context. Then the probabilities of the name type *EntityType* and *v*: *Prob(EntityType|v)* are compared. In addition, it roughly determines whether the name plays a subject or object role for *v*, based on simple heuristic rules using their relative positions and whether *v* is in a passive form. Finally two probability tables for subject/object are extracted separately. In total 67390 verbs are collected from a morphological dictionary automatically derived from COMLEX Syntax (Macleod et al., 1998). For example, in the subject table the statistics show:

Prob(GPE| "hire")= 0.029412, Prob(ORG| "hire")= 0.176471, Prob(PER | "hire")= 0

This indicates that when a name appears as the subject of "hire", it's more likely to be a GPE and ORG name instead of PER.

In contrast, in the object table:

Prob(PER| "hire")= 0.153846, Prob(ORG| "hire", object)= 0,

Prob(GPE, | "hire", object)= 0.

This indicates a PER name is very likely to appear as the object of "hire". These probabilities are scaled into ten bins and used as a re-ranking feature.

7.1.2.7 Coreference Features

Seven coreference features are captured for re-ranking the name hypotheses. For example, as mentioned in section 5.1.1.6, a name which is coreferred with more other mentions is more likely to be correct, so the number of mentions referring to a name candidate (*CorefNum*) is used.

One of the most common English errors is the ambiguity between name and nominal. If the baseline model cannot resolve this ambiguity, then it's possible that a nominal head will be mistakenly tagged as a name and at the same time get credit from the *CorefNum* feature. In order to alleviate this problem, *CorefNum* is skipped if the English name candidate *ncand* satisfies the following conditions:

((*ncand* is un-capitalized) and ((*ncand* includes single token) or (*ncand* is facility)))

or (*ncand* shares a head with other nominal mentions in the same entity)

The feature set also includes the weights of coreference links; for example, the link connecting two identical names will be assigned higher weight. For example, if the baseline tagger mistakenly identifies a name "American Council on Education" as "American Council", the coreference resolver using substring matching feature will still give credit to this name based on coreference with other mentions of "American Council on Education". But if different weights are assigned to capture the different coreference links, the re-ranker will prefer "American Council on Education" as a better name candidate. Other features are also encoded, such as how many other entities refer to the name candidates in the current hypothesis.

In order to incorporate wider context, cross-document coreference is applied for the test set. In particular, taking a cluster of documents about a single topic (e.g., reports from different sources or a sequence of successive reports from one source), the entities, relations, and events can be expected to be mentioned repeatedly.

The documents are clustered using a name-driven cross-entropy metric and then the entire cluster is treated as a single document. All the name candidates in the top *N* hypotheses are taken for each sentence in a cluster *T* to construct a "query set" *Q*. The metric used for the clustering is the cross entropy *H(T, d):*

$$H(T,d) = -\sum_{x \in Q} prob(T,x) \times \log prob(d,x),$$

where *prob(T, x)* is the distribution of the name candidate *x* in *T* , and *prob(T, x)* is the distribution of the name candidate *x* in document *d*. If *H(T, d)* is smaller than a threshold then we add *d* to *T*.

These clusters are built two ways: first, just clustering the test documents; second, by augmenting these clusters with related documents retrieved from a large unlabeled corpus (with document relevance measured using cross-entropy). Then we can process similar documents containing instances of the same name candidate, and combine the evidence from these additional instances with disambiguating contexts.

7.1.2.8 Conjunctions of Features

In addition to the individual features, selected conjunctions of related features are also included. For example, for a given margin confidence, certain ranking position of the candidate hypothesis are more likely to be better or worse than the baseline. So we can incorporate the conjunction of margin and ranking position features for the *BIRR* model (section 6.3.3.5) in order to capture the different impacts of margin value on candidate hypotheses.

Some other conjunction features of other pairs are encoded, such as (*$CorefNum_i$* - *$CorefNum_j$*) and (*$InRelation_i$* - *$InRelation_j$*), *HMMMargin* and (*$InRelation_i$* - *$InRelation_j$*) with intuitions shown in Table 5-1.

7.1.3 Case Study of Incremental Re-Ranking Framework on Chinese Name Tagging

The incremental re-ranking framework (section 6.4.5.4) is applied to MaxEnt-Rank for Chinese name tagging. A small additional gain is obtained by further splitting the first re-ranker into separate steps, with each step using the information from one subsequent stage. The overall architecture is presented in Figure 7-2.

The baseline name tagger generates N-Best multiple hypotheses for each sentence. Then the results from subsequent components are exploited in four incremental re-rankers. At each re-ranking stage, the information from one subsequent component is used, together with the probability score from the prior re-ranking stage. The high confidence hypotheses are generated as final output while low confidence hypotheses are sent through the next re-ranker.

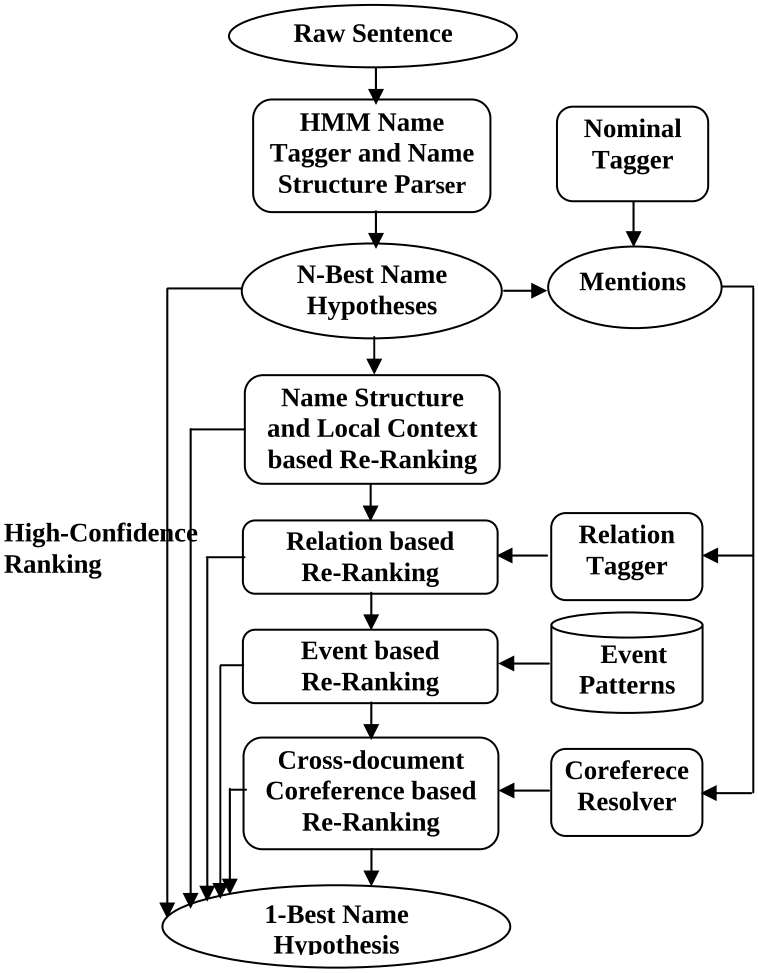


Figure 7-2. Incremental Name Re-Ranking Architecture

7.1.4 Experimental Results

7.1.4.1 Data and Experimental Setting

The re-ranking approach is tested for English and Chinese name tagging. Table 7-2 and Table 7-3 show the corpora used to train each stage and the test sets. Partial training (described in Figure 6-5) is used to generate training data for English name re-ranking, and K-Fold training for Chinese name re-ranking. The rule-based coreference resolver is used for the English experiment.

Component		Data
Training	Baseline name tagger	1375 texts from ACE 02, 03, 04, 05 training data
	Nominal tagger	English Penn TreeBank
	Relation tagger	328 ACE 04 texts
	Event pattern	A "verb-subject" probability table including 1537 entries and a 'verb-object' probability table including 1594 entries extracted from ACE04 training data using 3073 verbs in English PropBank
	Re-Ranker	23,674 samples from 310 texts from ACE 04 training data
Test		20 texts from ACE 04 training corpus, includes 743 names: 361 persons, 246 GPEs, 107 organizations, 12 locations, 13 facilities and 4 vehicles.

Table 7-2. English Data Description for Name Re-Ranking

Component		Data
Training	Baseline name tagger	2978 texts from the People's Daily in 1998 and 1300 texts from ACE 03, 04, 05 training data
	Nominal tagger	Chinese Penn TreeBank V5.1
	Coreference resolver	1300 texts from ACE 03, 04, 05 training data
	Relation tagger	633 ACE 05 texts, and 546 ACE 04 texts with types/subtypes mapped into 05 set
	Event pattern	376 trigger words, 661 patterns
	Name structure, coreference and relation based Re-Rankers	1,071,285 samples (pairs of hypotheses) from ACE 03, 04 and 05 training data
	Event based Re-Ranker	325,126 samples from ACE sentences including event trigger words
Test		100 texts from ACE 04 training corpus, includes 2813 names: 1126 persons, 712 GPEs, 785 organizations and 190 locations.

Table 7-3. Chinese Data Description for Name Re-Ranking

7.1.4.2 Overall Performance

The contributions of re-rankers in name identification and classification are evaluated separately, using MaxEnt-Rank(the OpenNLP package[8] is used), with BIRR function for English and PIRR function for Chinese as described in section 6.3.3.5. Tables 7-4 and 7-5 show the performance of Precision (P), Recall (R) and F-Measure (F) on identification, classification, and the combined task as re-rankers are added to the system.

Model	Identification			Classification Accuracy	Identification +Classification		
	P	R	F		P	R	F
Baseline	92.2	89.2	90.7	95.5	88.0	85.2	86.5
+Re-Ranking	92.4	91.1	91.7	95.7	88.4	87.2	87.8
Oracle (N=20)	93.9	97.3	95.6	99.0	93.0	96.4	94.6

Table 7-4. English Name Identification and Classification

Table 7-4 shows that re-ranking mainly helped to reduce English name identification missing errors. Although the overall gain in F-score is small (1.3%), the overall system achieves a 13.5% relative reduction on the missing rate over the baseline.

Model	Identification			Classification Accuracy	Identification +Classification		
	P	R	F		P	R	F
Baseline	93.2	93.4	93.3	93.8	87.4	87.6	87.5
+name structure	94.0	93.5	93.7	94.3	88.7	88.2	88.4
+relation	93.9	93.7	93.8	95.2	89.4	89.2	89.3
+event	94.1	93.8	93.9	95.7	90.1	89.8	89.9
+cross-doc coreference	95.1	93.9	94.5	96.5	91.8	90.6	91.2
Oracle (N=30)	97.6	98.8	98.2	98.6	96.2	97.4	96.8

Table 7-5. Chinese Name Identification and Classification

[8] http://maxent.sourceforge.net/index.html

Table 7-5 shows that the gain is greater for classification (2.7%) than for identification (1.2%) for Chinese name tagging. Furthermore, we can see that the gain in identification is produced primarily by the name structure and coreference components. As we noted earlier in section 5.1.1.3, the name structures can correct boundary errors by preferring names with complete internal components, while coreference can resolve a boundary ambiguity for one mention of a name if another mention is unambiguous. The greatest gains were therefore obtained in boundary errors: the stages together eliminated over 1/3 of boundary errors and about 10% of spurious names; only a few missing names were corrected, and some correct names were deleted.

Both relations and events contribute substantially to classification performance through their selectional constraints. The lesser contribution of events is related to their lower frequency.

Table 7-4 and 7-5 also show the oracle scores from this N-best / re-ranking strategy. Using the values of N (N<=20 or N<=30), the oracle F-measure is F=94.6% for English and F=96.8% for Chinese. This indicates that with relatively small values of N this approach is able to include highly-rated hypotheses for most sentences.

In order to check how robust the re-ranking approach is for name tagging, the Wilcoxon Matched-Pairs Signed-Ranks Test is conducted for both languages. For English we applied the test on each individual text, and for Chinese we split the test set into 10 folds, with 10 texts in each fold. The results show that the improvement using English re-ranking is significant at a 96.3% confidence level. For Chinese name re-ranking the improvements of gradually adding different feature types are significant at the following confidence levels: 97.1% for name structure features; 95.3% for relation features; 94.4% for event features; and 98.0% for cross-doc coreference features.

7.1.4.3 Impact of Cross-doc Coreference Features

As described in Section 7.1.2.7, these gains can be magnified by clustering documents and using cross-document coreference in these clusters. The 100 texts in the Chinese test set were first clustered into 28 topics (clusters). Then cross-document coreference is applied on each cluster. Compared to single document coreference, cross-document coreference obtained 0.5% higher F-Measure, with small gains in both identification (0.6% vs. 0.4%) and classification (0.8% vs. 0.4%), improving performance for 15 of these 28 clusters.

These clusters were then extended by selecting 84 additional related texts from a corpus of 15,000 unlabeled Chinese news articles (using a cross-entropy metric to select texts). 24 clusters gave further improvement, and an overall 0.2% further improvement on F-Measure was obtained.

7.1.4.4 Ranking Algorithm Comparison

Three re-ranking algorithms are evaluated: MaxEnt-Rank, SVMRank and p-norm Push Ranking described in section 6.3.4.7 on Chinese name tagging. $\text{SVM}^{\text{light}}$ (Joachims, 1998) is used for SVMRank, with a linear kernel and the soft margin parameter set to the default value. For the p-Norm Push Ranking, 33 weak rankers are applied. The number of iterations was fixed at 110, this number was chosen by optimizing the performance on a separate development set of 100 documents. Both MaxEnt-Rank and SVM-Rank use PIRR ranking function, and p-norm Push Ranking uses SDRR ranking function.

Model	P	R	F
Baseline	87.4	87.6	87.5
MaxEnt-Rank	91.8	90.6	91.2
SVMRank	89.5	90.1	89.8
p-Norm Push Ranking (p=16)	91.2	90.8	91.0

Table 7-6. Re-Ranking Algorithm Comparison for Chinese Name Tagging

Table 7-6 reports the overall performance for these three algorithms. All of them achieved improvements over the baseline. MaxEnt yields the highest precision, while p-Norm Push Ranking with p = 16 yields the highest recall. Therefore MaxEnt Rank and p-Norm Push Ranking proved about equally effective for this task.

7.1.5 Remaining Error Analysis

This section compares the remaining Chinese system errors with the human annotator's performance.

The use of 'feedback' from subsequent stages has yielded substantial improvements in Chinese name tagging accuracy, from F=87.5 with the baseline HMM to F=91.2. This performance compares quite favorably with the performance of the human annotators who prepared the ACE 2005 Chinese training data. The annotator scores (when measured against a final key produced by review and adjudication of the two annotations) were F=92.5 for one annotator and F=92.7 for the other. As in the case of the automatic tagger, human classification accuracy (97.2 - 97.6%) was better than identification accuracy (F = 95.0 - 95.2%).

Figure 7-3 summarizes the error rates for the baseline system, the improved system without coreference based re-ranker, the final system with re-ranking, and a single annotator.

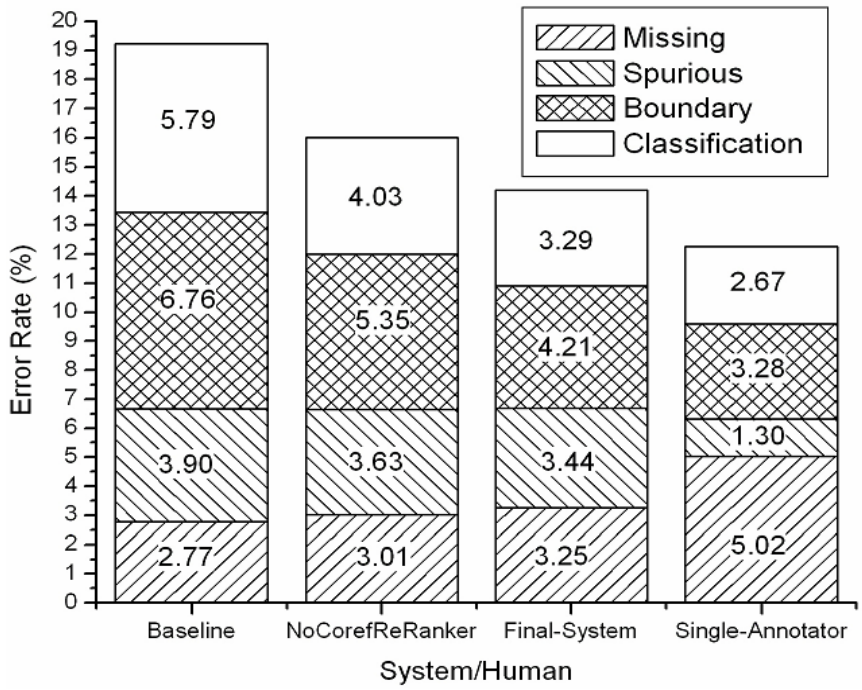


Figure 7-3. Chinese Name Error Distribution

Figure 7-3 shows that the performance improvement reflects a reduction in classification and boundary errors. Compared to the system, the human annotator's identification accuracy was much more skewed (52.3% missing, 13.5% spurious of all error types), suggesting that a major source of identification error was not difference in judgment but rather names which were simply overlooked by one annotator and picked up by the other.

Our analysis of the types of errors, and the performance of our knowledge sources, gives some indication of how further gains may be achieved. The selectional power of event extraction is limited by the frequency of event patterns – only about 1/3 of sentences had a pattern instance. Even with this limitation, a gain of 0.5% is obtained in name classification. Capturing a broader range of selectional patterns should yield further improvements. Nearly 70% of the spurious names remaining in the final output were in fact instances of 'other' types of names, such as book titles and building names; creating explicit models of such names should improve performance.

7.2 Improving Coreference Resolution by Relation Detection

This section will present another case study of cross-stage joint inference - using the output of the relation tagger to rescore coreference hypotheses and get the final coreference decisions.

7.2.1 Approach Overview

The framework of using relation features should, for example, be able to represent such basic interaction facts as whether the (possibly identical) people referenced by two mentions are in relations such as working in the same organization, or owning the same car, etc. Also it should be able to use this information to yield more accurate extraction results even when the local surface features at each individual stage are imperfect or fail altogether.

Here the pipeline consists of two stages, coreference resolution and relation detection. The baseline coreference resolver, as noted before in section 4.2.7, uses both absolute rules for ‘sure’ cases and a corpus-trained (MaxEnt) classifier for the remainder to produce a probability of coreference, without any relation information. Relation detection is applied next to acquire relation information. Particular coreference links are then *re-scored* (or, the ‘corefer’ and ‘not-corefer’ decisions are re-ranked), using both the coreference probability from the coreference resolver and features representing the relation knowledge sources.

7.2.2 Relation Feedback Features

This section will describe the features encoded from the results of relation detection to rescore coreference hypotheses.

Given the ACE-type relations, a semantic context for a candidate mention coreference pair (Mention 1b and Mention 2b) can be defined using a Relational Coreference Model (abbreviated as RCM) structure depicted in Figure 7-4.

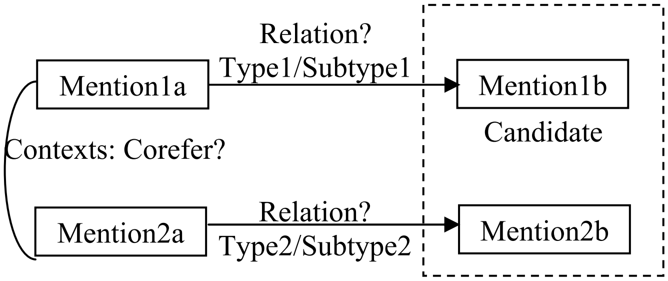


Figure 7-4. The Relational Coreference Model

If both mentions participate in relations, the model incorporates the types and directions of their respective relations as well as whether or not their relation partners (Mention 1a and Mention 2a) corefer. These values (which correspond to the edge labels in Figure 7-4) can then be factored into a coreference prediction. This RCM structure assimilates relation information into a coherent model of semantic context.

Any instance of the RCM structure needs to be converted into semantic knowledge that can be applied to a coreference decision. This problem is approached by constructing a set of RCM patterns and evaluating the accuracy of each pattern as positive or negative evidence for coreference. The resulting knowledge sources fall into two categories: rules that improve precision by pruning incorrect coreference links between mentions, and rules that improve recall by recovering missed links.

To formalize these relation patterns, based on Figure 7-4, the following clauses are defined:

A: RelationType1 = RelationType2

B: RelationSubType1 = RelationSubType2

C: Two Relations have the same direction

Same_Relation: $A \wedge B \wedge C$

CorefA: Mention1a and Mention2a corefer

CorefBMoreLikely: Mention1b and Mention2b are more likely to corefer

CorefBLessLikely: Mention1b and Mention2b are less likely to corefer

From these clauses the following plausible inferences are constructed:

Rule (7-1) $Same\_\mathrm{Re}lation \wedge \neg CorefA \Rightarrow CorefBLessLikely$

Rule (7-2) $\neg Same\_\mathrm{Re}lation \wedge CorefA \Rightarrow CorefBLessLikely$

Rule (7-3) $Same\_\mathrm{Re}lation \wedge CorefA \Rightarrow CorefBMoreLikely$

Rule (7-1) and (7-2) can be used to prune coreference links that simple string matching might incorrectly assert; and (7-3) can be used to recover missed mention pairs.

The accuracy of Rules (7-1) and (7-3) varies depending on the type and direction of the particular relation shared by the two noun phrases. For example, if Mention1a and Mention 2a both refer to the same nation, and Mentions 1b and 2b participate in citizenship relations (GPE-AFF) with Mentions 1a and 2a respectively, 1b and 2b should not necessarily refer to the same person. If 1a and 2a refer to the same person, however, and 1b and 2b are nations in citizenship relations with 1a and 2a, then it would indeed be the rare case in which 1b and 2b refer to two different nations. In other words, the relation of a nation to its citizens is one-to-many.

Our system learns broad restrictions like these by evaluating the accuracy of Rules (7-1) and (7-3) when they are instantiated with each possible relation type and direction and used as weak classifiers. For each such instantiation cross-validation is used on our training data to calculate a reliability weight defined as:

|Correct decisions by rule for given instance| /

#| Total applicable cases for given instance |

The number of correct decisions is counted for a rule instance by taking the rule instance as the only source of information for coreference resolution and making only those decisions suggested by the rule's implication (interpreting CorefBMoreLikely as an assertion that Mention 1b and Mention 2b do in fact corefer, and interpreting CorefBLessLikely as an assertion that they do not corefer).

Every rule instance with a reliability weight of 70% or greater is retained for inclusion in the final system. Rule (7-2) cannot be instantiated with a single type because it requires that the two relation types be different, and so this filtering is not performed for Rule (7-2) (Rule (7-2) has 97% accuracy across all relation types).

This procedure yields 58 reliable (reliability weight > 70%) type instantiations of Rule (7-1) and (7-3), in addition to the reliable Rule (7-2). An additional 24 reliable rules can be recovered by conjoining additional boolean tests to less reliable rules. Tests include equality of mention heads, substring matching, absence of temporal key words such as "current" and "former," number agreement, and high confidence for original coreference decisions (Mention1b and Mention2b). For each rule below the reliability threshold, the algorithm searches for combinations of 3 or fewer of these restrictions until achieving reliability of 70% or the search space has been exhausted.

For example, the following two instantiations are considered as reliable: Two cities located in different countries are very unlikely to corefer ("PHYS/Located"); Two organizations producing the same airplane are very likely to corefer ("Artifact/Owner").

If a high reliability instantiation of one of these RCM rules applies to a given mention-antecedent pair, the following features are included for that pair: the type of the RCM rule, the reliability of the rule instantiation, the relation type and subtype, the direction of the relation, and the tokens for the two mentions.

7.2.3 Experimental Results

7.2.3.1 Data and Scoring

The system is evaluated on English and Chinese. Table 7-7 and 7-8 summarize the training corpora and blind test sets used for the components in these two languages.

Component		Data
Training	Baseline coreference resolver	311 newswire and newspaper texts from the ACE 2002 and ACE 2003 training corpora
	Relation tagger	328 ACE 04 texts
	Re-Scorer	126 ACE 04 newswire texts
Test		65 ACE 04 newswire texts

Table 7-7. English Data Description for Coreference Re-Scoring

Component		Data
Training	Baseline coreference resolver	767 texts from ACE 2003 and ACE 2004 training corpora
	Relation tagger	646 ACE 04 texts
	Re-Scorer	646 ACE 04 texts
Test		100 ACE 04 texts

Table 7-8. Chinese Data Description for Coreference Re-Scoring

The MUC coreference scoring metric (Vilain et al., 1995) is used to evaluate our systems. This metric uses the ACE keys and only scores mentions which appear in both the key and system response.

7.2.3.2 Overall Performance

We performed experiments to evaluate the impact of coreference rescoring using relation information. Table 7-9 lists the results.

Model	Performance	Recall	Precision	F-measure
English	Baseline	77.2	87.3	81.9
	Rescoring	80.3	87.5	83.7
Chinese	Baseline	75.0	76.3	75.6
	Rescoring	76.1	76.5	76.3

Table 7-9. Performance of Re-Scoring for Coreference Resolution

Table 7-9 shows that the relation information provided some improvements for both languages. Relation information increased both recall and precision in both cases.

A sign test applied to a 5-way split of each of the test corpora indicated that for both languages, the system that exploited relation information significantly outperformed the baseline (at the 95% confidence level, judged by F-measure).

7.2.4 Conclusion

This section 7.2 has outlined an approach to improving coreference resolution through the use of semantic relations, and has described a system which can exploit these semantic relations effectively. The experiments on English and Chinese data showed that these small inroads into semantic territory do indeed offer performance improvements.

8. CASE STUDY ON CROSS-LINGUAL INTERACTION

This chapter will present an example of cross-task interactions: combining the quite disparate knowledge sources from source language entity extraction and entity translation to produce better results for each. In contrast to the case studies presented in Chapter 7 which used only mono-lingual interactions, this chapter will focus on incorporating cross-lingual feedback. In the experiments presented in this chapter, the source language is Chinese and target language is English.

8.1 Interaction between Entity Extraction and Entity Translation

Section 5.2 presented examples that indicate how aligned bitexts can aid entity extraction. Huang and Vogel (2002) used these observations to improve the name tagging of a bitext, and the NE (named entity) dictionary learned from the bitext. However, in most cases the texts from which to extract entities will not be part of such bitexts. This section aims to use information which can be gleaned from bitexts to improve the tagging of data for which we do not have pre-existing parallel text.

Therefore this thesis will instead use a phrase-based statistical machine translation system which is trained from these bitexts and thus in effect distills the knowledge in its training bitexts; the source-language entities will be translated using the entity translation system described in section 4.2.8; and then this translation will be used to improve the entity extraction of the original text, as the bitext examples shown in section 5.2. A new framework is proposed to incorporate the properties of the translated entities as follows.

Firstly a source language ‘baseline’ entity extraction system is applied to produce entities (*SEntities*), and then these entities are translated into target language *T (TEntities)*. Coreference decisions are made on the source language level. The *TEntities* carry information from a machine translation system trained from large bitexts, information which may not have been captured in the monolingual entity extraction. The *TEntities* can be used to provide *cross-lingual feedback* to confirm the results or repair the errors in *SEntities*. A set of rules incorporating such feedback are applied iteratively[9].

However, the translations produced by the MT system will not always be correct. This thesis addresses this problem by using confidence estimation based on voting among translations of coreferring mentions, which will be referred to as a *mention cache*.

Section 8.2.1 will verify the two hypotheses which are required to apply the cache scheme, and Section 8.2.2 will explain the details of these caches.

8.2 Cross-lingual Voted Caches

In the following, section 8.2.1 will present two hypotheses required to apply the cross-lingual voted caches, and then section 8.2.2 will describe the implementation details.

8.2.1 Hypotheses

8.2.1.1 One Translation per Named Entity

Named entities may have many variants, for example, “IOC” and “International Olympic Committee” refer to the same entity; and “New York City” alternates with “New York”;

[9] In the Chinese-English experiments presented in this chapter, the correction procedure converges after 3-4 loops.

but all these different variants tend to preserve ‘name heads’ – a brief “key” alternation that represents the *naming function* (Carroll, 1985). Unlike common words for which *fluency* and *vitality* are most required during translation, translating a named entity requires preserving its *functional* property – the real-world object that the name is referring to. Inspired by this linguistic property, a hypothesis is proposed as follows:

- **Hypothesis (8-1).** *One Translation per Named Entity:*

The translation of different name mentions is highly consistent within an entity.

This hypothesis may seem intuitive, but it is important to verify its accuracy. On 50 English documents (4360 mention pairs) from ACE 2007 Chinese to English Entity Translation training data with human tagged entities, this hypothesis’ *accuracy* is measured by computing:

$$\frac{\#\,|\text{ Coreferred mention pairs with } \textit{consistent} \text{ translations }|}{\#\,|\text{Coreferred mention pairs }|}$$

Two translations are considered as *consistent* if one is a name component (e.g “Kensington Land” = “Kensington”), acronym (e.g “European Union” = “EU”) or adjective form of the other (e.g “Iraqi = Iraq”).

The *accuracy* of this hypothesis for different name types are: 99.6% for PER, 99.5% for GPE, 99.0% for ORG and 100% for LOC. This clearly indicates that Hypothesis (1) holds with high reliability.

8.2.1.2 One Source Name per Translation

Based on Hypothesis (8-1), a single ‘best (maximal) name translation’ can be selected for each entity with a name; and this best translation can be used as ‘*feedback*’ to determine whether the extracted name mentions in source language are correct or not. If they are incorrect (if their translations are not consistent with the best translation), they can be replaced by a ‘best source language name’. This is justified by:

- **Hypothesis (8-2).** *One Source Name per Translation*:

Names that have the same translation tend to exhibit *consistent* spellings in the source language.

In reviewing 101 Chinese documents (8931 mention pairs) with human translations from ACE07 entity translation training data, the accuracy of this hypothesis for all entity types was close to 100%; the exceptions appeared to be clear translation errors.

Therefore, if the name mentions in one entity are required to achieve consistent translation as well as extraction (name boundary and type), then the within-doc or cross-doc entity-level errors can be fixed with small sacrifice of (<1%) exceptional instances.

8.2.2 Implementation

Given an entity in source language *SEntity* and its translation *TEntity*, let *SName(i)* be a name mention of *SEntity* with translation *TName(i)*. Then the above two properties indicate that if string *TName(i)* appears frequently in *TEntity*, then *SName(i)* is likely to be correct. On the other hand, if *TName(i)* is infrequent in *TEntity* and conflicts with the most frequent translation in boundary or word morphology, then *SName(i)* is likely to be a wrong extraction.

For a pair of languages *S* (source language) → *T* (target language), the following voted cache models are built in order to get the best *assignment* (extraction or translation candidate) for each entity:

- ***Inside-S-T-Cache***

For the names of one entity (inside a single document), record their unique translations and frequencies;

- ***Cross-S-T-Cache***

Corpus-wide (across documents), for each name and its consistent variants, record its unique translations and their frequencies;

- ***Cross-T-S- Cache***

Corpus-wide, for each set of consistent name translations in T, record the corresponding names in S and their frequencies.

These caches are depicted in Figure 8-1. They incorporate simple filters based on properties of language *T* to exclude translations which are not likely to be names. For *T* = English, the following translations are excluded: empty translations, translations which are single un-capitalized tokens, and, for person names, translations with any un-capitalized tokens. In addition, in counting translations in the cache, consistent translations are grouped together. For English, this includes combining person name translations if one is a subsequence of the tokens in the other. The goal of these simple heuristics is to take advantage of the general properties of language *T* in order to increase the likelihood that the most frequent entry in the cache is indeed the best translation.

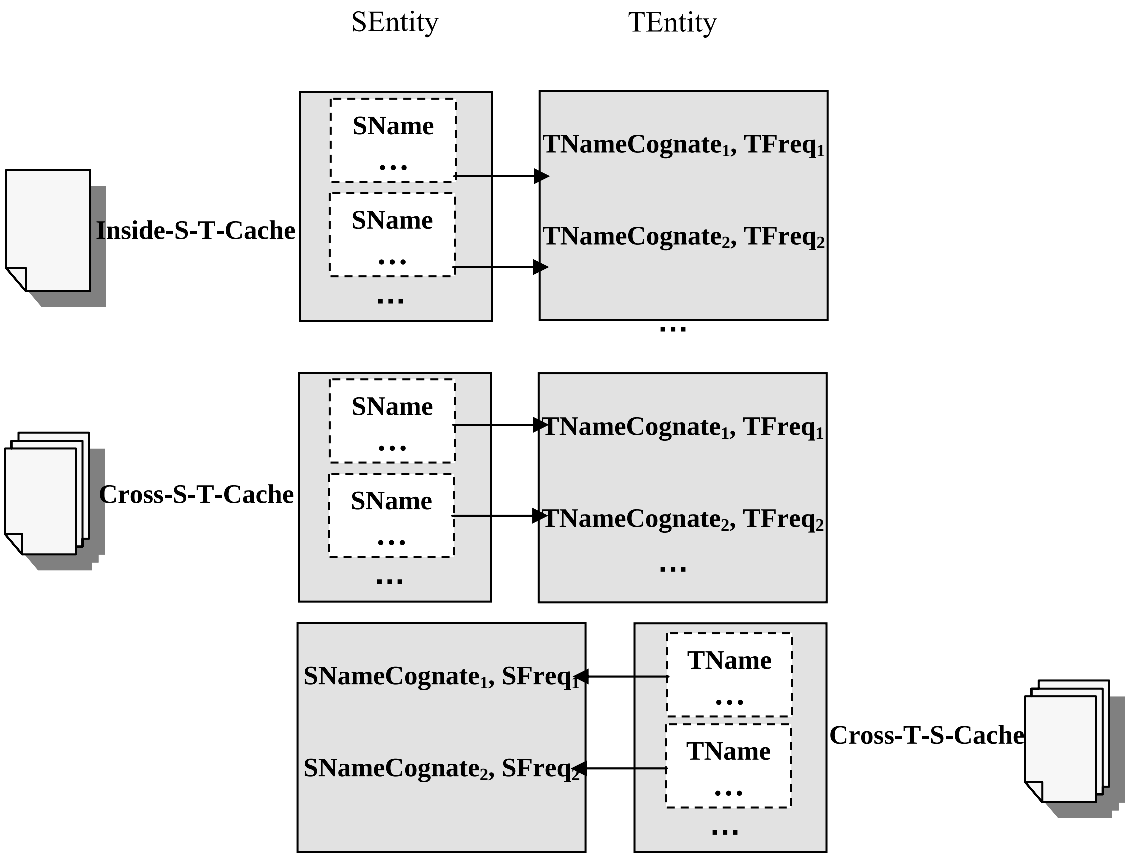


Figure 8-1. Cross-lingual Voted Caches

For each entry in these caches, the frequency of each unique *assignment* is counted, and then the following *Cross-lingual Margin (CMargin)* measurement is used to compute the confidence of the best assignment:

CMargin = Frequency (Best Assignment) –Frequency (Second Best Assignment)

A large CMargin indicates greater confidence in the assignment. The value of CMargin will be incorporated into inference rules in the next section.

The algorithm identifies name coreference relations in the source language and compares the extractions and translations of coreferring mentions, applying these voting caches operating over source and machine-translated entities. The source language coreference information is thus incorporated as a confidence metric and the high-confidence best assignment (extraction or translation) can be propagated through the corpus, replacing lower-confidence assignments. The detailed inference rules will be described in the next section.

8.3 Inference Rules

The language-specific information in *SEntity* and its entry in the cross-lingual caches can be combined to detect potential extraction errors and take corresponding corrective measures.

Based on hypotheses (8-1) and (8-2), for a test corpus a group of entities is produced in both source and target languages, with high consistency on the following levels:

Rule (8-1): Adjust Source Language Annotations to Achieve Mention-level Consistency:

Rule (8-1-1): Adjust Mention Identification

If a mention receives a translation that has a small CMargin as defined in section 8.2.2 and violates the linguistic constraints in target language, then classify the mention as a nominal or discard it from the mention set.

Rule (8-1-2): Adjust Isolated Mention Boundary

Adjust the boundary of each mention of *SEntity* to be consistent with the mention receiving the best translation.

Rule (8-1-3): Adjust Adjacent Mention Boundary

If two adjacent mentions receive the same translation with high confidence, merge them into one single mention.

Rule (8-2): Adjust Source Language Annotations to Achieve Entity-level Consistency:

If one entity is translated into two groups of different mentions, split it into two entities.

Rule (8-3): Adjust Target Language Annotations to Achieve Mention-level Consistency:

Enforce entity-level translation consistency by propagating the high-confidence best translation through coreferred mentions.

These inferences are formalized in Table 8-1 and exemplified in Table 8-2. These rules are tested on a development set, and a few source-language-specific restrictions on their applicability were added to improve performance. Also, where the rules allowed for two alternative corrections, a language-specific criterion is added for choosing the correction. Specifically: for Rule (8-1-2) also checks that *SName(i)* and *SName(j)* are not a name and its acronym. Also for Rule (8-1-2), if *SName(i)* includes a conjunction the rule splits the name into two names, otherwise replacing it by *SName (j)*. For Rule (8-1-1), since in Chinese most ambiguities between name and nominal arise in GPE or ORG names, GPE or ORG names are corrected into nominals, while PER names are deleted. Rule (8-1-3) was limited to merging mentions of selected entity type pairs, such as "PER-GPE" and "ORG-LOC" because they are unlikely to appear adjacent in Chinese.

<table>
<tr><th colspan="2">Terms</th></tr>
<tr><td><i>TConstraint</i></td><td>Some constraint that name entities must satisfy in language <i>T</i>. For example, in the setting of <i>S</i>=Chinese and <i>T</i>=English, it includes the capitalization constraint.</td></tr>
<tr><td><i>CorefMentionNum(i)</i></td><td>the number of name mentions coreferring to <i>SName(i)</i> in <i>SEntity</i></td></tr>
<tr><td><i>BestTName(Cache)</i></td><td>the best (most frequent) translation in <i>Cache</i></td></tr>
<tr><td><i>FreBestTName(Cache)</i></td><td>the frequency of the best (most frequent) translation in <i>Cache</i></td></tr>
<tr><td><i>FreSeBestTName(Cache)</i></td><td>the frequency of the second best (most frequent) translation in <i>Cache</i></td></tr>
<tr><td><i>CMargin(i,Cache)</i></td><td>the C<i>Margin</i> (defined in section 8.2.2) of name <i>SName(i)</i> in <i>Cache</i></td></tr>
<tr><th colspan="2">Predicates</th></tr>
<tr><td><i>ViolateTConstraint(i)</i></td><td><i>TName(i)</i> does not satisfy <i>TConstraint</i></td></tr>
<tr><td><i>HasBestTran(j, Cache)</i></td><td><i>SName(j)</i> has translation <i>BestTName(Cache)</i> in <i>Cache</i></td></tr>
<tr><td><i>ConflictBoundary(i, j)</i></td><td><i>SName(i)</i> is consistent with <i>SName(j)</i> at one boundary but not the other</td></tr>
<tr><td><i>HasFewCorefMentions(i)</i></td><td><i>CorefMentionNum(i)</i> &lt; δ_1</td></tr>
<tr><td><i>HasLowConf(i, Cache)</i></td><td><i>CMargin(i, Cache)</i> &lt; δ_2</td></tr>
<tr><td><i>ShareTranslation(i, j)</i></td><td><i>TName(i) = TName(j)</i></td></tr>
<tr><td><i>Adjacent(i, j)</i></td><td><i>SName(i)</i> and <i>SName(j)</i> are adjacent to each other</td></tr>
<tr><td><i>EqualConf(SEntity)</i></td><td><i>FreBestTName(Inside-S-T-Cache)</i> > δ_3 ∧
<i>FreSeBestTName(Inside-S-T-Cache)</i> > δ_4</td></tr>
<tr><td><i>Overlap(i, j)</i></td><td><i>SName(i) and SName(j)</i> overlap in spelling</td></tr>
<tr><th colspan="2">Rule (8-1-1): Adjust Mention Identification</th></tr>
<tr><td colspan="2">if <i>(ViolateTConstraint(i)</i> ∧ (<i>HasFewCorefMentions(i)</i> ∨ <i>HasLowConf(i, Cross-T-S-Cache)))</i> then
Change <i>SName(i)</i> into nominal or delete it</td></tr>
<tr><th colspan="2">Rule (8-1-2): Adjust Isolated Mention Boundary</th></tr>
<tr><td colspan="2">for all j ≠ i do
if <i>(ViolateTConstraint(i)</i> ∧ <i>HasBestTran(j, Inside-S-T-Cache)</i> ∧ <i>ConflictBoundary(i, j))</i> ∨
<i>(HasBestTran(j, Cross-T-S-Cache)</i> ∧ <i>ConflictBoundary(i, j))</i> then
Replace <i>SName(i)</i> with <i>SName(j)</i> or split it into <i>SName(j)</i> and another mention</td></tr>
<tr><th colspan="2">Rule (8-1-3): Adjust Adjacent Mention Boundary</th></tr>
<tr><td colspan="2">for all j ≠ i do
if <i>ShareTranslation(i, j)</i> ∧ <i>Adjacent(i, j)</i> then Merge <i>SName(i)</i> and <i>SName(j)</i> into a single mention</td></tr>
<tr><th colspan="2">Rule (8-2): Adjust Entity-level Consistent Source Language Annotation (Coreference Resolution)</th></tr>
<tr><td colspan="2">if <i>EqualConf(SEntity)</i> ∧ ¬<i>Overlap(i, j)</i> then Split <i>SEntity</i> into two entities</td></tr>
<tr><th colspan="2">Rule (8-3): Adjust Mention-level Consistent Target Language Annotation (Mention Translation)</th></tr>
<tr><td colspan="2">if ¬<i>HasLowConf(i, Inside-S-T-Cache)</i> then Replace <i>TName(i)</i> with <i>BestTName(Inside-S-T-Cache)</i>
if ¬<i>HasLowConf(i, Cross-S-T-Cache)</i> then Replace <i>TName(i)</i> with <i>BestTName(Cross-S-T-Cache)</i></td></tr>
</table>

Table 8-1. Inference Rules of Using Translation to Improve SEntity Extraction

These rules are applied repeatedly until there are no further changes; improved translation in one iteration can lead to improved *S* entity extraction in a subsequent iteration. For example, for the following Chinese document,

<TEXT>
<sent 1>加拿大第 37 届联邦议会 29 日举行会议，选举自由党议员***米利肯***为众议院新议长。</sent>
The 37th Canadian Federal Parliament held a meeting on the 29th and elected Liberal MP ***Miliken*** as House of Commons speaker.

<sent2>今年 54 岁的***彼得.米利肯***是来自加拿大安大略省金斯敦地区的议员。</sent>
The 54-year-old ***Peter Miliken*** is a MP from Kingston, Ontario, Canada.

<sent3>***米利肯***是在 5 轮投票后当选的。</sent>
Miliken was elected after five rounds of voting.
</TEXT>

The baseline system extracts and translates the following entity:

{米利/Mili, 彼得.米利肯/ Peter Miliken, 米利肯/ Miliken}

By applying rule (8-1-2), the boundary of the first name mention “*米利*” can be fixed into “*米利肯*” because “*米利肯*” has the (maximal) best translation “*Miliken*”:

{米利肯/Mili,彼得.米利肯/Peter Miliken, 米利肯/Miliken}

then by applying rule (8-3) the translation “Mili” can be changed into the more frequent translation “*Miliken*” [10]:

{米利肯/Miliken,彼得.米利肯/Peter Miliken, 米利肯/Miliken}

More examples are presented in Table 8-2.

[10] Alternatively we could fix the English in this case by re-translating the corrected mentions. But in other cases rule (3) is needed to correct the translations.

Rule	Improvement	Baseline		After Using Inference Rule	
		SEntities	**TEntities**	**SEntities**	**TEntities**
8-1-1	**Name Identification**	***总联盟*** ***联盟***	*general union* *alliance*		
8-1-2	**Isolated Name Boundary**	*阿贾比* ***阿贾比由*** *哈米德.阿贾比* *阿贾比*	*Agabi* ***Agabi from*** *Hamid Agabi* *Agabi*	*阿贾比* ***阿贾比*** *哈米德.阿贾比* *阿贾比*	*Agabi* ***Agabi*** *Hamid Agabi* *Agabi*
8-1-3	**Adjacent Name Boundary**	***乌兹*** ***别克斯坦***	***Uzbekistan*** ***Uzbekistan***	***乌兹别克斯坦***	***Uzbekistan***
8-2	**Coreference Resolution**	***埃及约旦*** *约旦*	***Egypt and Jordan*** *Jordan*	***约旦*** *约旦* ***埃及***	***Jordan*** *Jordan* ***Egypt***
8-3	**Name Translation**	*以色列* *以色列* *以色列* *以* *以*	*Israel* *Israel* *Israel* ***as***	*以色列* *以色列* *以色列* *以* *以*	*Israel* *Israel* *Israel* ***Israel*** ***Israel***

Table 8-2. Example for Applying Cross-lingual Inference Rules

8.4 Architecture Overview

The overall system pipeline for language pair (*S, T*) is summarized in Figure 8-2.

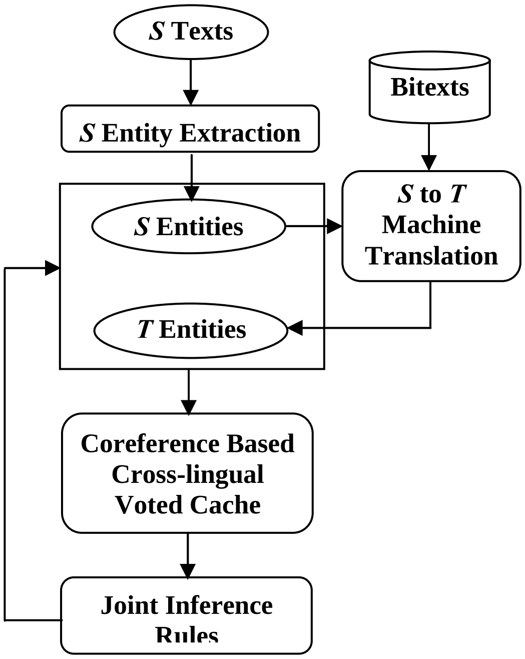


Figure 8-2. A Symbiotic Framework of Entity Extraction and Translation

8.5 Experimental Results

This section will report present the experiments on using Chinese-to-English translation to improve Chinese entity extraction.

8.5.1 Data

The Chinese newswire data is taken from the ACE 2007 Entity Translation training and evaluation corpus and used as our blind test set. The test set includes 67 news texts, with 2077 name mentions and 1907 entities. A separate development set including 100 news texts was used to develop the inference rules.

8.5.2 Improvement in Entity Extraction

The name tagging performance on different entity types is shown in Table 8-3 as follows.

Type	Baseline	After Using Inference Rules
PER	89.9%	91.2%
GPE	87.0%	86.9%
ORG	85.7%	88.5%
LOC	89.7%	90.6%
FAC	80.9%	85.3%
ALL	87.3%	89.2%

Table 8-3. Improvement in Source Name Tagging F-measure

Except for the small loss for GPE names, our method achieved positive corrections on most entity types (2.2% relative improvement in name tagging F-measure, representing a 15.0% error reduction).

Significant improvements were achieved on ORG and FAC names, mainly because organization and facility names in English texts have less boundary ambiguity than in Chinese texts. So they are better aligned in bitexts and easier to translate. The small loss in GPE names for the Chinese source is due to the poor quality of the translation of country name abbreviations.

The rules can also improve nominal tagging by disambiguating mention types (name vs. nominal) and improve coreference by merging or splitting incorrect entity structures. All of these improvements benefit entity extraction. To get a sense of the overall performance of our method on entity extraction, Table 8-4 shows the results with the official ACE EDR Value metric[11].

[11] The description of the ACE entity extraction (EDR) metric can be found at: http://www.nist.gov/speech/tests/ace/ace07/doc/ace07-evalplan.v1.2.pdf

Type	Baseline	After Using Inference Rules
PER	70.8	71.4
GPE	73.6	74.4
ORG	60.2	63.3
LOC	43.9	43.6
FAC	51.8	52.0
ALL	66.6	67.9

Table 8-4. Improvement in Source Entity Extraction

Improvements in entity value substantially exceeding those for name F-measure reflect gains in nominal tagging and coreference. For example, GPE entities had worse name tagging but better coreference. LOC had better performance in name tagging but got worse ACE value, because some spurious name mentions were mistakenly changed to nominal mentions. This suggests employing a more effective supervised learning approach to assign weights and applying priorities for various inference rules.

The improved mentions can be propagated to the entity level by re-running the Chinese coreference resolver at the end. By doing this further gains of 0.3 are obtained in ACE entity value. In other experiments, this method was found to achieve more gains for automatic speech recognition (ASR) transcripts than regular newswire texts. Transcripts usually don't include sentence-internal punctuation, so adjacent mentions will be more likely to be mistakenly merged into one single mention. After applying rules (8-1-1), (8-1-2) and (8-1-3) to split these mentions, coreference resolution can be applied again to assign correct coreference links to the new mentions.

8.5.3 Improvement in Entity Translation

A further benefit of our system is a boost in the translation quality of Chinese entities. The official ACE 2007-ET scorer[12] is used to measure the F-scores. The performance for translating different entity types is presented in Table 8-5.

Table 8-5 shows that this approach achieved absolute improvement on entity translation (9.1% relative improvement in F-measure). The inference based on voting over mentions of an entity particularly improved translation for GPE abbreviation names such as "叙 (Syria)", "拉美 (Latin America)" which otherwise will be missed or mistakenly translated into non-names, and fixed translated person foreign name boundaries. It also successfully translates more non-famous names because of the global propagation of correct assignments. Thus the method has used the interaction of entity extraction and translation to improve the performance of both.

Type	Baseline	After Using Inference Rules
PER	34.8%	36.7%
GPE	44.7%	49.8%
ORG	37.0%	39.9%
LOC	18.3%	18.1%
FAC	23.1%	23.3%
ALL	35.1%	38.3%

Table 8-5. Improvement in Entity Translation

[12] The description of the ACE entity translation metric can be found online at http://www.nist.gov/speech/tests/ace/ace07/doc/ET07-evalplan-v1.6.pdf

8.5.4 Error Analysis

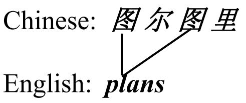
The errors (cases where cross-lingual inference rules made entity extraction worse) reveal both the shortcomings of the entity translation system and consistent difficulties across languages.

For a name not seen in training bitexts the MT system tends to mistakenly align part of the name with an un-capitalized token. For example,

(a) Results from Chinese entity tagger

*一位发言人说，由25至30人组成的一群武装分子星期三在特里普拉邦首府阿加塔拉以南60公里（38英里）的<ENAMEX TYPE="GPE">**图尔图里**</ENAMEX>，袭击了村里江湖医生的家。*

(b) Bitext


Chinese: **图尔图里**

English: ***plans***

(c) Entity Tagging after using bitext

*一位发言人说，由25至30人组成的一群武装分子星期三在特里普拉邦首府阿加塔拉以南60公里（38英里）的<NOMINAL>**图尔图里**</NOMINAL>，袭击了村里江湖医生的家。*

The GPE name "***图尔图里***" ("Tuertuli") was correctly identified by the baseline name tagger, while it's not in the bitext used to train the MT system; parts of it (the first and third character) were mistakenly aligned to an un-capitalized token "plans" in English. Also, there are words where the ambiguity between name and nominal exists in both Chinese and English, such as "国会–parliament". Rule (8-1-1) fails in these cases by mistakenly changing correct names into nominal mentions. In these and other cases, a separate GPE name transliteration system could be developed from larger name-specific bitexts to re-translate these difficult names. Or the pipeline could incorporate the confidence values such as (Ueffing and Ney, 2005) generated from the MT system into our cross-lingual cache model.

8.6 Conclusion

Bitexts can provide a valuable additional source of information for improving entity extraction. The above sections have demonstrated how the information from bitexts, as captured by an entity translation system, and then used to generate translations, can be used to correct errors made by a source-language entity extraction baseline. Such bitexts, between a language *S* and *T*, are now being harvested in fairly large quantities - much larger than we could afford to annotate by hand. Note that this approach does not assume that we have bitexts for the data we need to name-tag, only that there exist bitexts for data in the same general domain (and involving some of the same names).

The work described in this chapter complements the research described by (Huang and Vogel, 2002). Unlike their approach requiring reference translations in order to achieve highest alignment probability, only the source language unlabeled document is needed. So this approach is more broadly applicable and also can be extended to additional information extraction tasks (nominal tagging and coreference).

In place of correction rules, a joint inference approach can be adopted to generate alternative source language name tags (with probabilities), estimate the probabilities of the corresponding target language features, and seek an optimal tag assignment. Although the current approach only relies on limited target language features, a full target-language entity extractor (as Huang and Vogel (2002) did) could be used to provide more information as feedback (for example, name type information).

9. RELEVANT WORK

Most of the work in this thesis has been published over the past three years; these publications are listed in Appendix A. Over this time a variety of related research has cited these publications; some of these papers are listed below.

9.1 Joint Inference between Mention Detection and Coreference Resolution

Joint Inference between mention detection and coreference resolution has become a topic of keen interest. However, most researchers used a quite different framework from our approaches.

Zhou et al. (2005) applied Transformation Based Learning to incorporate the feedback from coreference resolution as rules to improve mention detection; they reported a 1.7% absolute improvement on ACE EDR value.

Daume III (2006) proposed structured prediction algorithms for mention detection and coreference based on complex, non-local interaction features.

Poon and Domingos (2007) used a Markov logic model consisting of logical formulas representing the interactions, to improve candidate mentions (citation records) and coreference resolution simultaneously.

The idea of using the number of coreferring mentions for pruning errant names was applied to cross-document person name normalization in (Magdy et al., 2007).

Very recently, Vilain et al. (2007) compared five name taggers, and demonstrated that the key issue is ensuring the tagging coherence at the whole-document level. This might help alleviate error propagation with a dual-pass strategy that particularly afflicts long documents. The re-ranking approach presented in the thesis is one promising way to address the problem.

9.2 Joint Inference between Name Tagging and Relation Detection

(Yangarber and Jokipii, 2005) presented an information correction system, in which multiple hypotheses of related names were extracted from a large document collection; then these hypotheses were propagated to system outputs, and the corrected results were used as 'feedback' to back-propagate to repair components that induced incorrect information.

9.3 Using Semantic Features for Coreference Resolution

Since 2005 researchers in coreference resolution area have returned to the once-popular semantic-knowledge-rich approach, investigating a variety of semantic knowledge sources. (Ponzetto and Strube, 2005a; 2005b) used the semantic relationships between a predicate and its arguments as document-level constraints to improve coreference resolution. They also incorporated other semantic knowledge sources from WordNet and Wikipedia as features. They improved 8.9% F-measure for broadcast news and 2.7% for newswire.

Ng (2007) automatically acquired semantic class knowledge from a version of the Penn Treebank with semantic classes labeled. Two features from the induced semantic classes were used: whether two mentions have the same semantic class, and whether they belong to some particular semantic classes. These constraints provided 2% improvement in F-measure.

Jing et al. (2007) applied the joint inference rule (7-2) presented in Chapter 7 as constraints in coreference clustering. These constraints provided significant improvement over the IBM coreference resolver based on lexical and syntactic features (Luo at al. 2004, Luo and Zitouni, 2005), especially for noisy mention inputs (broadcast conversation in their paper). They also benefited from this symbiosis framework between coreference and relations to extract a social networks and biographies for conversational transcripts.

9.4 Re-Ranking for NLP

In the incremental re-ranking framework proposed in Chapter 6, it's assumed possible to generate the structurally correct solutions incrementally, through a sequence of partially completed solutions. Ginter et al. (2006) employed this idea and extended it to a new algorithm that aimed to identify the globally best solution, without fully completing all structurally correct solutions.

Carvalho and Cohen (2007) applied the Classification based Direct Re-Ranking algorithm as described in Chapter 6 into the task of email recipient recommendation.

9.5 Software Application

Our Chinese name tagger is freely available for research purposes. There have been some successful applications in other NLP areas such as soundbite speaker name recognition (Liu and Liu, 2007a) and language modeling adaptation (Liu and Liu, 2007b).

10. CONCLUSIONS AND FUTURE WORK

This chapter concludes the dissertation by summarizing joint inference work and proposing several directions for future work.

10.1 Conclusion

The key characteristic of this thesis is its role in solving the bottlenecks of two traditional IE frameworks: sequential and monolithic. The thesis proposed a new IE framework based on stage interactions. This thesis has presented the detailed interaction features and a variety of re-ranking algorithms to implement this framework. The effectiveness of this framework has been demonstrated by three case studies: the interaction between coreference resolution/relation detection/event detection and name tagging; relation detection and coreference resolution; source language entity extraction and entity translation. These improvements are encouraging when one considers that these are only part of a larger set of interactions in a NLP pipeline which we can explore.

10.2 Future Work

This section proposes some other possible intriguing interactions which can be further explored.

10.2.1 More Mono-lingual Interactions

First, it should be possible to exploit more interactions between different IE stages.

10.2.1.1 Interaction between Chinese Word Segmentation and Name Tagging

The current Chinese baseline name tagger is based on words instead of characters. (Jing et al., 2003) showed that a character-based HMM outperforms the word-based model by 3-5.5% F-measure. Comparable gains may not be achieved because the Chinese word segmenter used in this thesis doesn't have a granularity problem as mentioned by Jing et al. – the segmenter tends to segment unknown words (including name candidates) into individual characters. Therefore it has already approached in the direction of character-based HMM.

Despite this the name re-ranking can be extended to a character-based model, keeping multiple segmentations among the N-Best name hypotheses. When the best name hypothesis for each sentence is selected, the best segmentation result can be produced simultaneously. Ultimately the subsequent IE stages such as chunking can also benefit from the improved word segmentation results.

10.2.1.2 Interaction with Nominal Tagging

ACE nominal mention detection may be viewed as a specialized form of word sense disambiguation (WSD). We can, of course, use traditional WSD methods (based on a statistical analysis of context words) to address this task. However, it's also possible to take advantage of the interactions with other stages, as for name identification and classification. The preference for coherent discourse suggests that isolated mentions (with no coreference or semantic links) are more likely to be errors, so semantic class assignments which license such links should be preferred. Experiments may need to be conducted to compare the effectiveness of these approaches against traditional WSD methods.

10.2.1.3 Interaction between Coreference Resolution and Event Detection

- **Within-document Coreference Resolution and Event Detection**

Section 5.1.2.2 showed that event detection results can be used as feedback to correct coreference resolution. In the current Chinese coreference resolver, this evidence is encoded as additional rules, for example, for any two entity mentions M_1, M_2 appearing as arguments in a event mention,

(1) For Business/Life/Movement/Conflict/Contact/Justice/Transaction/Personnel-End-Position events, if M_1, M_2 are not in an apposition, they are unlikely to be coreferential; (2) For Personnel (Start-Position, Nominate, Elect) events, if M_1 and M_2 are persons, they are likely to be coreferential (for example, “Fred” and “president” will be coreferential in “Fred was named president.”). But more specific constraints need to be encoded as re-scoring features. It’s also expected that parallel event structures can aid coreference much the way parallel relation structures do.

- **Cross-document Coreference Resolution and Event Detection**

ACE events don’t appear often in the texts, but their benefits for coreference can be magnified by looking across documents. For example, if “the Palestinian” in document1 and “Abbas” in document2 are involved in the same “attack” event contexts, then it’s more likely they are referring to each other. On the other hand, if “Halid Siehl Mohammed” and “Mohammed” were “arrested” in two different places at different times, then they are unlikely to refer to the same entity. Such event-oriented cross-document coreference results can be further used to fill in missed arguments or correct wrong arguments for the relevant events.

The intuition is clear, but more systematic studies need to be conducted to determine what type of event clustering is most effective. Some event-driven document clustering approaches such as the relevant sentence sets returned by an information retrieval engine can be options as pre-processing for getting candidate event clusters. Another problem for this idea is the difficulty of constructing key data for evaluation. Such data may be derived from existing reference data for question-answering tasks, or even evaluate the task indirectly based on the performance of its applications.

10.2.2 More Cross-Task Joint Inference

This approach could be applied more broadly, to different NLP tasks. There are a number of natural extensions and generalizations of cross-task interactions.

10.2.2.1 Interaction between Source Entity Extraction and Target Entity Extraction

The entity extraction and translation work described in Chapter 8 can be extended to cross-lingual bootstrapping.

Another alternative approach is to directly operate on MT training data (sentence aligned bilingual corpus), using the interactions between source and target entity extraction. The source and target entity extractors can run in parallel with confidence estimation, and then for each pair of sentences <$SSent_i$, $TSent_i$>, the entity type annotations can be used together with confidence values to correct them based on annotation consistency. Using this entity-corrected bilingual corpus, more accurate entity extractors can then be re-trained for each language. This procedure can be repeated until we get a bilingual corpus with satisfactory entity annotations and better entity extractors.

10.2.2.2 Interaction between Source Event Extraction and Target Event Extraction

Similarly, we can incorporate the interaction between source and target language event extractors to improve each. For example, if the Chinese event extractor is not sure of associating the pair of <*STrigger, SArgument$_i$*>, but if the English event extractor has high confidence at identifying <*TTrigger, TArgument$_i$*>, where *STrigger-TTrigger* and *SArgument$_i$-TArgument$_i$* are aligned by a MT system, then the likelihood of <*STrigger, SArgument$_i$*> can be increased correspondingly. A preliminary test on 19 ACE Chinese texts achieved 2.9 more ACE event value over a low baseline.

Also the MT training data can be used in the same way as described in section 10.2.2.1 to obtain more event training data for the two event extractors.

10.2.2.3 Interaction between Speech Recognition and Information Extraction/Translation

Recently there has been rapid progress in applying text processing techniques on the output of automatic speech recognition (ASR) (Makhoul et al., 2005). The potential ASR transcription errors, in particular name spelling errors, make IE more difficult. It’s possible to use text processing results as feedback to achieve document-level consistency and correct these errors. For example,

Example 10-1. Using Coreference Resolution Feedback to Correct ASR error

<Sent1>*嗯哼我们看到这个在中东的反应哈马斯方面当然就是马上就显得这么一个裁决也当年呢****萨姆****还是给一些钱呢*</Sent>

<Sent2>*其实你现在****萨达姆****虽然犯下那么多的这些*</Sent>

<Sent3>*他们一向是这样说而且****萨达姆***</Sent>

If the coreference resolver can cluster the three names “***萨姆***’, “***萨达姆***’ and “***萨达姆***’ into one entity based on substring, then it may be possible to recover the first name “***萨姆***’ into “***萨达姆***’.

Example 10-2. Using Name Translation Feedback to Correct ASR error

<Sent1>*再看的是正在墨西哥访问的加州州长****施瓦辛格****周四就指出民主党在美国中期选举得*</Sent>

<Sent2>***施瓦辛格****周四和墨西哥总统福克斯共进早餐*</Sent>

<Sent3>*那么共和党籍的****史瓦辛格****也认为其他的共和党人也可以学习他和民主党合作的经验*</Sent>

<Sent4>*而****施瓦辛格****会在墨西哥逗留两天*</Sent>

The name “Schwarzenegger” appears four times in the text (“***施瓦辛格***’, “***施瓦辛格***’, “***史瓦辛格***’ and “***施瓦辛格***’). They have the same pronunciations but the instance in sentence 3 “***史瓦辛格***” has wrong spelling. If the name translation component can successfully translate them into the correct English name “Schwarzenegger”, it’s possible to correct “***史瓦辛格***’ into “***施瓦辛格***’.

Besides using the feedback knowledge as inference rules to correct such errors, they can also be incorporated to expand ASR vocabulary and conduct correction iteratively.

10.2.2.4 Interaction between Speech Sentence Segmentation and Information Extraction

Some automatic sentence segmentation and comma prediction systems (Zimmermann et al., 2006; Hillard et al., 2006) have been developed to assist better IE on ASR output. IE results may be used as feedback to select the best segmentation hypothesis. The central idea is to contain each individual entity mention, relation mention or event mention (including trigger and arguments) within one sentence. Some potential correction examples are presented as follows (manually translated Mandarin data).

Example 10-3. Using Name Tagging to Correct Speech Sentence Segmentation error

Speech Sentence Segmentation:

<Sent1>*Decide to continue protecting Chen Shuibian and contacting with* ***Democracy***</Sent>

<Sent2>***Progress Party****'s active members</Sent>*

After Using Name Tagging Feedback:

<Sent1>*Decide to continue protecting Chen Shuibian and contacting with* ***Democracy Progress Party****'s active members</Sent>*

Example 10-4. Using Nominal Detection to Correct Speech Sentence Segmentation error

Speech Sentence Segmentation:

<Sent1>*Uh all the committees and departments are now creating* ***their own*** </Sent>

<Sent2>***governmental websites<****/Sent>*

After Using Nominal Mention Detection Feedback:

<Sent1> *Uh all the committees and departments are now creating* ***their own***
governmental websites <*/Sent>*

Example 10-5. Using Event Detection to Correct Speech Sentence Segmentation error

Speech Sentence Segmentation:

<Sent1>*Uh* ***Iraqi security agency*** *members were very likely to have*

been</Sent>

<Sent2>***meeting*** *with the* ***Armed Force<****/Sent>*

After Using Event Detection Feedback:

<Sent1>*Uh* ***Iraqi security agency*** *members were very likely to have been*
meeting *with the* ***Armed Force<****/Sent>*

Appendix A. RELEVANT PUBLICATIONS

Some of the material in this thesis has appeared in the proceedings of natural language processing conferences. The first of our papers that proposed the idea of joint inference between IE stages appeared at an ACL 2004 workshop:

> Heng Ji and Ralph Grishman. 2004. Applying Coreference to Improve Name Recognition. *Proc. ACL 2004 Workshop on Reference Resolution and Its Applications.* pp. 32-39. Barcelona, Spain.

In ACL 2005 we extended this idea and proposed the joint inference framework for the whole IE pipeline, and used name tagging as a case study. Besides coreference information we also incorporated semantic relations as feedback features. Another improvement was to incorporate all the interaction knowledge into a statistical re-ranking model:

> Heng Ji and Ralph Grishman. 2005. Improving Name Tagging by Reference Resolution and Relation Detection. *Proc. ACL 2005*. pp. 411-418. Ann Arbor, USA.

Then we applied the joint inference framework to another case study on improving coreference resolution using the feedback from relation detection. And we demonstrated the idea on two languages – English and Chinese. The paper appeared at HLT/EMNLP 2005:

> Heng Ji, David Westbrook and Ralph Grishman. 2005. Using Semantic Relations to Refine Coreference Decisions. *Proc. HLT/EMNLP 2005*. pp. 17-24. Vancouver, B.C., Canada

During the ACE 2005 evaluation we extensively explored many possible interactions among IE stages. For example, we incorporated event patterns to improve name tagging. The ACE 2005 system incorporated the joint inference methods described in the thesis:

Heng Ji, Adam Meyers and Ralph Grishman. 2005. NYU's Chinese ACE 2005 EDR System Description. *Proc. ACE 2005 Evaluation/PI Workshop*. Washington, US.

Till the end of 2005 we tried incorporating some kinds of global information, and extended the feedback from coreference to the cross-document level. So it became necessary to analyze the remaining error types for name tagging. We divided the errors into different identification and classification types, and analyzed how joint inference helped to solve them. We also took a further step by studying the remaining errors that global features didn't repair, and compared the results with a single human annotator. These results were published at COLING/ACL 2006:

Heng Ji and Ralph Grishman. 2006. Analysis and Repair of Name Tagger Errors. *Proc. COLING/ACL 2006*. Sydney, Australia.

We believe it's also important to do research on the aspect of learning models for joint inference. During 2006 spring we applied a brand-new ranking algorithm "p-Norm Push Ranking" to implement joint inference for name tagging. Then we compared the results with two other re-ranking models: MaxEnt-Rank and SVMRank. The results were presented in the paper:

Heng Ji, Cynthia Rudin and Ralph Grishman. 2006. Re-Ranking Algorithms for Name Tagging. *Proc. HLT/NAACL 06 Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing.* New York, NY, USA.

During the fall of 2006, we tried to take a further step of applying joint inference to a cross-lingual IE system. We decided to look outside of IE, and use the outputs of machine translation as feedback to enhance source language name tagging and coreference resolution. By doing this we benefited from the bitexts (bilingual corpora and lists) that were indirectly incorporated in machine translation. This collaborative model was prestend at RANLP 2007:

> Heng Ji and Ralph Grishman. 2007. Collaborative Entity Extraction and Translation. *Proc. International Conferences on Recent Advances in Natural Language Processing 2007*. Borovets, Bulgaria.

BIBLIOGRAPHY

David Bean and Ellen Riloff. 2004. Unsupervised Learning of Contextual Role Knowledge for Coreference Resolution. *Proc. of the HLT-NAACL2004*. pp. 297-304. Boston, USA.

Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance Learning Name-finder. *Proc. of the Fifth Conf. on Applied Natural Language Processing*. pp.194-201. Washington D.C., USA.

E. Black, F. Jelinek, J. Lafferty, Magerman D. M., R. Mercer, and S. Roukos. 1993. Towards history-based grammars: Using richer models for probabilistic parsing. *Proc. of the ACL 1993*. pp. 31-37.

Andrew Borthwick. 1999. *A Maximum Entropy Approach to Named Entity Recognition*. Ph.D. Dissertation, Dept. of Computer Science, New York University.

Elizabeth Boschee, Ralph Weischedel and Alex Zamanian. 2005. Automatic Evidence Extraction. *Proc. of the International Conference on Intelligence Analysis*. McLean, VA.

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R.L. Mercer and P. S. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2): 79-85.

Jaime Carbonell and Ralf Brown. 1988. Anaphora resolution: A multi-strategy approach. *Proc. of the COLING 1988*, pp.96-101

Xavier Carreras and Lluis Marquez. 2004. Introduction to the CONLL-2004 Shared Task: Semantic Role Labeling. *Proc. of the CONLL 2004*.

Xavier Carreras and Lluis Marquez. 2005. Introduction to the CONLL-2005 Shared Task: Semantic Role Labeling. *Proc. of the CONLL 2005*.

John M. Carroll. 1985. *What's in a Name?: An Essay in the Psychology of Reference*. New York, US.

Victor R. Carvalho and William W. Cohen. Recommending Recipients in the Enron Email Corpus. To appear. http://www.cs.cmu.edu/~wcohen/postscript/cc-predict-submitted.pdf

Eugene Charniak. 1972. *Toward a model of children's story comprehension*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Eugene Charniak. 1973. Jack and Janet in Search of a Theory of Knowledge. *Proc. Of the Advance Papers from the Third International Joint Conference on Artificial Intelligence*, Stanford, CA, US. pp. 337-343.

Eugene Charniak. 2001. Unsupervised Learning of Name Structure From Coreference Data. *Proc. of the NAACL 01*. pp. 48-54.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-Fine N-Best Parsing and MaxEnt Discriminative Reranking. *Proc. of the ACL2005*. pp. 173-180. Ann Arbor, USA

John Chen, Srinivas Bangalore, Michael Collins and Owen Rambow. 2002. Reranking an n-gram supertagger. *Proc. of the Sixth International Workshop on Tree Adjoining Grammars and Related Frameworks.*

David Chiang. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. *Proc. of the ACL2005*. pp. 263-270. Ann Arbor, USA

Hai Leong Chieu and Hwee Tou Ng. 2002. Named Entity Recognition: A Maximum Entropy Approach Using Global Information. *Proc. of the COLING 2002*, Taipei, Taiwan.

Yejin Choi, Eric Breck and Claire Cardie. 2006. Joint Extraction of Entities and Relations for Opnion Recognition. *Proc. of the EMNLP 2006*. pp. 431-439.

Yen-Lu Chow and Richard Schwartz. 1989. The N-Best Algorithm: An efficient Procedure for Finding Top N Sentence Hypotheses. *Proc. of the DARPA Speech and Natural Language Workshop*. pp. 199-202

Michael Collins. 2002. Ranking Algorithms for Named-Entity Extraction: Boosting and the Voted Perceptron. *Proc. of the ACL 2002*. pp. 489-496

Michael Collins and Nigel Duffy. 2002. New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. *Proc. of the ACL2002*. pp. 263-270. Philadelphia, USA.

Michael Collins and Terry Koo. 2003. Discriminative Reranking for Natural Language Parsing. *Journal of Association for Computational Linguistics*. pp. 175-182

K. Crammer and Y. Singer. 2001. PRanking with Ranking. *Proc. of the Advances in Neural Information Processing Systems (NIPS)*. pp. 641-647

N. Cristianini, J. Schawe-Taylor. 2000. *An Introduction to Support Vector Machines*. Cambridge University Press.

Harold Charles Daume III. 2006. *Practical Structured Learning Techniques for Natural Language Processing.* Ph.D. Dissertation, Dept. of Computer Science, University of Southern California.

Harold Charles Daume III and Daniel Marcu. 2005. A Large-Scale Exploration of Effective Global Features for a Joint Entity Detection and Tracking Model. *Proc. of the HLT/EMNLP 2005*. Vancouver, Canada.

David Eppstein. 2001. *K shortest paths and other "K Best" problems*. http:/www.ics.uci.edu/~eppstein/bibs/kpath.bib

Jenny Rose Finkel, Trond Grenager and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proc. of the ACL 2005*. pp. 363-370.

Radu Florian, Hany Hassan, Hongyan Jing, Nanda Kambhatla, Xiaoqiang Luo, Nicolas Nicolov and Salim Roukos. 2004. A Statistical Model for Multilingual Entity Detection and Tracking. *Proc. of the HLT-NAACL 2004*. Boston, Mass, USA.

Radu Florian, Hongyan Jing, Nanda Kambhatla and Imed Zitouni. 2006. Factorizing Complex Models: A Case Study in Mention Detection. *Proc. of the COLING-ACL 2006*, pp. 473-480. Sydney, Australia.

Radu Florian, Ding-Jung Han, Xiaoqiang Luo, Nanda Kambhatla and Imed Zitouni. 2007. IBM ACE'07 System Description. *Proc. of the ACE 2007 PI/Evaluation Workshop*.

Yoav Freund, Raj Iyer, Robert E. Schapire and Yoram Singer. 1998. An efficient boosting algorithm for combining preferences. *Machine Learning: Proceedings of the Fifteenth International Conference*. pp. 170-178.

Yoav Freund and Robert E. Schapire. 1999. Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, 37(3): 277-296.

Yoav Freund, Raj Iyer, Robert E. Schapire and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4. pp.933-969

Jianfeng Gao, Mu Li, Andi Wu and Chang-Ning Huang. 2005. Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach. *Computational Linguistics*, 31(4). pp. 531-574.

Niyu Ge, John Hale and Eugene Charniak. 1998. A statistical approach to anaphora resolution. *Proc. of the Sixth Workshop on Very Large Corpora*.

Filip Ginter, Aleksandr Myllari and Tapio Salakoski. 2006. Probabilistic Search for the Best Solution Among Partially Completed Candidates. *Proc. of the HLT/NAACL 06 Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*. New York, NY, USA.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference - 6: A brief history. *Proc. of the 16th Int'l Conference on Computational Linguistics (COLING 96)*, Copenhagen. pp. 466-471

Ralph Grishman. 2004. NYU’s English ACE EDR & RDR system. *Proc. of the ACE Evaluation/PI Workshop*, September 2004, Alexandria, VA.

Ralph Grishman, David Westbrook and Adam Meyers. 2005. NYU’s English ACE 2005 System Description. *Proc. of the ACE 2005 Evaluation/PI Workshop*.

Michael A. K. Halliday and Ruqaiya Hasan. 1976. Cohesion in English. Longman.

Sanda M. Harabagiu, Razvan C. Bunescu and Steven J. Maiorano. 2001. Text and Knowledge Mining for Coreference Resolution. *Proc. of the NAACL 2001.* pp. 55-62.

James Henderson and Ivan Titov. 2005. Data-Defined Kernels for Parse Reranking Derived from Probabilistic Models. *Proc. of the ACL 2005*. pp. 181-188. Ann Arbor, USA.

R. Herbrich, T. Graepel, and K. Obermayer. 2000. Large margin rank boundaries for ordinal regression. In A.J. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pp. 115-132. MIT Press.

Dustin Hillard, Zhongqiang Huang, Heng Ji, Ralph Grishman, Dilek Hakkani-Tur, Mary Harper, Mari Ostendorf, Wen Wang. Impact of Automatic Comma Prediction on POS/Name Tagging of Speech. *Proc. of the IEEE/ACL 2006 Workshop on Spoken Language Technology*, Aruba, December 2006.

Jerry Hobbs, Mark Stickel, Douglas Appelt and Paul Martin. 1993. Interpretation as abduction. *Artificial Intelligence*, 63. pp. 69-142.

Jerry Hobbs. 1985. *On the Coherence and Structure of Discourse*. CLSI Technical Report 85-37. Stanford, CA, USA.

Fei Huang and Stephan Vogel. 2002. Improved Named Entity Translation and Bilingual Named Entity Extraction. *Proc. of the ICMI 2002*, pp. 253-258. Pittsburgh, PA, US.

Zhongqiang Huang, Mary Harper and Wen Wang. 2007. Mandarin Part-Of-Speech Tagging

and Discriminative Reranking. *Proc. of the EMNLP 2007*. Prague, Czech Republic.

Heng Ji, Matthias Blume, Dayne Freitag, Ralph Grishman, Shahram Khadivi and Richard Zens. 2007. NYU-Fair Isaac-RWTH Chinese to English Entity Translation 07 System. *Proc. of the ACE ET 2007 PI/Evaluation Workshop.* Maryland, US.

Heng Ji and Ralph Grishman. 2004. Applying Coreference to Improve Name Recognition. *Proc. of the ACL 2004 Workshop on Reference Resolution and Its Applications*. pp. 32-39. Barcelona, Spain.

Heng Ji and Ralph Grishman. 2005. Improving Name Tagging by Reference Resolution and Relation Detection. *Proc. of the ACL2005*. pp. 411-418. Ann Arbor, USA.

Heng Ji and Ralph Grishman. 2006a. Data Selection in Semi-supervised Learning for Name Tagging. *Proc. of the COLING/ACL 2006 Workshop on Information Extraction Beyond the Document*, pp.48-55. Sydney, Australia.

Heng Ji and Ralph Grishman. 2006b. Analysis and Repair of Name Tagger Errors. *Proc. of the COLING/ACL 2006*. Sydney, Australia.

Heng Ji and Ralph Grishman. 2007. Collaborative Entity Extraction and Translation. *Proc. of the RANLP 2007*. Borovets, Bulgaria. Sept 2007.

Heng Ji, Adam Meyers and Ralph Grishman. 2005. NYU's Chinese ACE 2005 EDR System Description. *Proc. of the ACE 2005 Evaluation/PI Workshop*. Washington, US.

Heng Ji, Cynthia Rudin and Ralph Grishman. 2006. Re-Ranking Algorithms for Name Tagging. *Proc. of the HLT/NAACL 06 Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*. New York, NY, USA.

Heng Ji, David Westbrook and Ralph Grishman. 2005. Using Semantic Relations to Refine Coreference Decisions. *Proc. of the HLT/EMNLP2005*. pp. 17-24. Vancouver, B.C., Canada

Hongyan Jing, Radu Florian, Xiaoqiang Luo, Tong Zhang and Abraham Ittycheriah. 2003. HowtogetaChineseName(Entity): Segmentation and Combination Issues. *Proc. of the EMNLP2003*.

Hongyan Jing, Nanda Kambhatla and Salim Roukos. 2007. Extracting Social Networks and Biographical Facts From Conversational Speech Transcripts. *Proc. of the ACL 2007*, pp. 1040-1047. Prague, Czech Republic.

Thorsten Joachims. 1998. Making large-scale support vector machine learning practical. *Advances in Kernel Methods: Support Vector Machine*. MIT Press.

Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. *Proc. of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*.

Taku Kudo, Jun Suzuki and Hideki Isozaki. 2005. Boosting-based Parse Reranking Derived from Probabilistic Models. *Proc. of the ACL2005*, pp. 189-196. Ann Arbor, USA.

John Lafferty, Andrew McCallum and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proc. of the 18th International Conference on Machine Learning*. San Francisco, CA, USA.

Feifan Liu and Yang Liu. 2007a. Look Who is Talking: Soundbite Speaker Name Recognition in Broadcast News Speech. *Proc. of the HLT/NAACL 2007.* pp. 101-104. Rochester, NY, USA.

Feifan Liu and Yang Liu. 2007b. Unsupervised Language Model Adaptation Incorporating Named Entity Information. *Proc. of the ACL 2007.* pp. 672-679. Prague, Czech Republic.

Xiaoqiang Luo and Imed Zitouni. 2005. Multi-Lingual Coreference Resolution With Syntactic Features. *Proc. of the HLT/EMNLP 2005*. pp. 660-667. Vancouver, Canada.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. *Proc. of the ACL 2004*. pp. 135-142. Barcelona, Spain.

Catherine Macleod, Ralph Grishman and Adam Meyers. 1998. COMPLEX Syntax. *Computers and the Humanities.* Volume 31. pp. 459-481.

Walid Magdy, Kareem Darwish, Ossama Emam and Hany Hassan. 2007. Arabic Cross-Documnt Person Name Normalization. *Proc. of the ACL 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources.*

John Makhoul, Alex Baron, Ivan Bulyko, Long Nguyen, Lance Ranshaw, David Stallard, Richard Schwartz and Bing Xiang. The Effects of Speech Recognition and Punctuation on Information Extraction Performance. *Proc. of the Interspeech 2005*. pp. 57-60.

Katja Markert and Malvina Nissim. 2005. Comparing Knowledge Sources for Nominal Anaphora Resolution. *Computational Linguistics*, 31(3). pp. 367-401.

Andrew McCallum and Wei Li. 2003. Early results for Named Entity Recognition With Conditional Random Fields, Feature Induction, and Web-Enhanced Lexicons. *Proc. of the CONLL-2003*, Edmonton, Canada.

Adam Meyers, Michiko Kosaka, Satoshi Sekine, Ralph Grishman and Shubin Zhao. 2001. Parsing and GLARFing. *Proc. of the RANLP 2001*.

Scott Miller, Jethran Guinness and Alex Zamanian. 2004. Name Tagging with Word Clusters and Discriminative Training. *Proc. of the HLT/NAACL 2004*, pp. 337-342. Boston, Massachusetts, US.

Ruslan Mitkov. 2001. Towards a more consistent and comprehensive evaluation of anaphora resolution algorithms and systems. *Applied Artificial Intelligence*, Vol. 15, Number 3, 2001, pp. 253-276.

Mehryar Mohri and Michael Riley. 2002. An Efficient Algorithm for the N-Best-Strings Problem. *Proc. of the ICSLP '02*, pp. 1313-1316.

Alessandro Moschitti, Daniele Pighin and Roberto Basili. 2006. Semantic Role Labeling via Tree Kernel Joint Inference. *Proc. of the CONLL 2006*.

Hwee Tou Ng, Bin Wang, and Yee Seng Chan. 2003. Exploiting Parallel Texts for Word Sense Disambiguation: An Empirical Study. *Proc. of the ACL 2003*, pp. 455-462. Sapporo, Japan.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. *Proc. of the ACL 2002*, pp.104-111.

Vincent Ng. 2007. Semantic Class Induction and Coreference Resolution. *Proc. of the ACL 2007*. pp. 536-543. Prague, Czech Republic.

F. J. Och and Hermann Ney. 2002. Discriminative training and Maximum Entropy Models for statistical Machine Translation. *Proc. of the ACL 2002*.

F. J. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. *Proc. of the ACL 2003*.

John Platt, Nello Cristianini, and John Shawe-Taylor. 2000. Large margin dags for multiclass classification. *Advances in Neural Information Processing Systems 12*, pp. 547-553.

Simone Paolo Ponzetto and Michael Strube. 2006a. Semantic Role Labeling for Coreference Resolution. *Proc. of the EACL 2006*. pp. 143-146.

Simone Paolo Ponzetto and Michael Strube. 2006b. Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. *Proc. of the HLT/NAACL 2006*. pp. 192-199.

Hoifung Poon and Pedro Domingos. 2007. Joint Inference in Information Extraction. *Proc. of the AAAI07.*

Dan Roth and Wen-tau Yih. 2002. Probabilistic Reasoning for Entity & Relation Recognition. *Proc. of the COLING2002*. pp. 835-841.

Dan Roth and Wen-tau Yih. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. *Proc. CONLL 2004*. pp. 1-8.

Dan Roth and Wen-tau Yih. 2007. Global Inference for Entities and Relations Identification via a Linear Programming Formulation. To appear in *Statistical Relational Learning*, Getoor and Taskar Edits.

Cynthia Rudin, Corinna Cortes, Mehryar Mohri, Robert E. Schapire. 2005. Margin-Based Ranking and Boosting Meet in the Middle. *Proc. of the 18th Annual Conference on Learning Theory (COLT'05).*

Cynthia Rudin. 2006. Ranking with a p-Norm Push. *Proc. of the Nineteenth Annual Conference on Computational Learning Theory (COLT 2006)*, Pittsburgh, Pennsylvania, US.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active Hidden Markov Models for Information Extraction. *Proc. of the International Symposium on Intelligent Data Analysis (IDA-2001)*.

Satoshi Sekine, Ralph Grishman and Hiroyuki Shinnou. 1998. A Decision Tree Method for Finding and Classifying Names in Japanese Texts. *Proc. of the Sixth Workshop on Very Large Corpora*; Montreal, Canada.

Satoshi Sekine and Chikashi Nobata. 2004. Definition, Dictionary and Tagger for Extended Named Entities. Proc. LREC 2004.

Libin Shen and Aravind K. Joshi. 2003. An SVM Based Voting Algorithm with Application to Parse ReRanking. *Proc. of the HLT-NAACL 2003 workshop on Analysis of Geographic References*. pp. 9-16.

Libin Shen and Aravind K. Joshi. 2004. Flexible Margin Selection for Reranking with Full Pairwise Samples. *Proc. of the IJCNLP2004*. pp. 446-455. Hainan Island, China.

Libin Shen, Anoop Sarkar and Franz Josef Och. 2004. Discriminative Reranking for Machine Translation. *Proc. of the NAACL 04*, pp. 177-184.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, Volume 27, Number 4, pp. 521-544

Jian Sun, Jianfeng Gao, Lei Zhang, Ming Zhou and Changning Huang. 2002. Chinese Named Entity Identification Using Class-based Language Model. *Proc. of theCOLING 2002*.

Mihai Surdeanu, Sanda Harabagiu, John Williams and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. *Proc. of the ACL2003*. Sapporo, Japan.

Koichi Takeuchi and Nigel Collier. 2002. Use of Support Vector Machines in Extended Named Entity Recognition. *Proc. of the CONLL 2002*. Taipei, Taiwan.

Joel R. Tetreault. 2001. A corpus-based evaluation of centering and pronoun resolution. Computational Linguistics, Volume 27, Number 4, pp. 507-520

Joel R. Tetreault and James Allen. 2004. Semantics, Dialogue, and Pronoun Resolution. *Proc. of the CATALOG '04*. Barcelona, Spain.

Kristina Toutanova, Aria Haghighi and Christopher D. Manning. 2005. Joint Learning Improves Semantic Role Labeling. *Proc. of the ACL 2005*.

Nicola Ueffing and Hermann Ney. 2005. Word-Level Confidence Estimation for Machine Translation using Phrase-Based Translation Models. *Proc. of the HLT/EMNLP 2005*, pp.763-770. Vancouver, Canada.

Renata Vieira and Massimo Poesio. An Empirically-based System for Processing Definite Descriptions. *Computational Linguistics*, 26(4): pp. 539-593.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. *Proc. the 6th Message Understanding Conference (MUC-6)*. San Mateo, Cal. Morgan Kaufmann.

Marc Vilain, Jennifer Su and Suzi Lubar. 2007. Entity Extraction is a Boring Solved Problem – or is it? *Proc. of the HLT/NAACL 2007*. pp. 181-184.

Min Wan and Zhensheng Luo. 2003. Study on Topic Segmentation Method in Automatic Abstracting System. *Proc. of the Natural Language Processing and Knowledge Engineering*, pp. 734-739. Oct. 2003.

Tuangthong Wattarujeekrit. 2005. *Exploring Semantic Roles for Named Entity Recognition in the Molecular Biology Domain*. PhD Dissertation. Japanese National Institute of Informatics.

Ben Wellner, Andrew McCallum, Fuchun Peng and Michael Hay. 2004. An Integrated, Conditional Model of Information Extraction and Coreference with Application to Citation Matching. *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004.

Robert Wilensky. 1983. *Planning and Understanding*. Addison-Wesley.

Terry Winograd. 1972. *Understanding Natural Language*. Academic Press. New York.

Nianwen Xue and Martha Palmer. 2003. Annotating Propositions in the Penn Chinese Treebank. *Proc. of the 2nd SIGHAN Workshop on Chinese Language Processing*. Sapporo, Japan.

Xiaofeng Yang, Guodong Zhou, Jian Su and Chew Lim Tan. 2003. Coreference Resolution Using Competition Learning Approach. *Proc. of the ACL 2003*.

Xiaofeng Yang, Jian Su and Chew Lim Tan. 2006. Kernel-Based Pronoun Resolution with Structured Syntactic Knowledge. *Proc. of the COLING/ACL 2006*, pp. 41-48.

Roman Yangarber and Lauri Jokipii. 2005. Redundancy-based Correction of Automatically Extracted Facts. *Proc. of the HLT/EMNLP 2005*, pp. 57-64.

Shiren Ye, Tat-Seng Chua, Liu Jimin. 2002. An Agent-based Approach to Chinese Named Entity Recognition. *Proc. of the COLING 2002*.

Dmitry Zelenko, Chinatsu Aone, and Jason Tibbets. 2004. Binary Integer Programming for Information Extraction. *Proc. of the ACE Evaluation Meeting*, September 2004, Alexandria, VA.

Richard Zens and Hermann Ney. 2004. Improvements in Phrase-Based Statistical Machine Translation. *Proc. of the HLT/NAACL 2004*. New York City, NY, US.

Lufeng Zhai, Pascale Fung, Richard Schwartz, Marine Carpuat, and Dekai Wu. 2004. Using N-best Lists for Named Entity Recognition from Chinese Speech. *Proc. of the NAACL 2004 (Short Papers)*, pp. 37-40.

Yaqian Zhou, Changning Huang, Jianfeng Gao and Lide Wu. 2005. Transformation Based Chinese Entity Detection and Tracking. *Proc. of the IJCNLP 2005.*

Matthias Zimmermann, Dilek Hakkani-Tür, James Fung, Nikki Mirghafori, Luke Gottlieb, Yang Liu, and Elizaneth Shriberg. The ICSI+ Multi-Lingual Sentence Segmentation System. *Proc.of the Interspeech 2006*. Pittsburgh, PA. September 2006.